**PROPOSAL**

**PERBANDINGAN ARSITEKTUR CNN (MOBILENETV1 DAN**

**RESNET18) DALAM KLASIFIKASI BARANG RUMAH**

**UNTUK ANAK USIA DINI BERBASIS ANDROID**

**Diajukan Untuk Memenuhi**

**Salah Satu Syarat Memperoleh Gelar Sarjana Teknik**



**MUHAMMAD ISHAK JAELANI**

**E1E120042**

**JURUSAN TEKNIK INFORMATIKA**

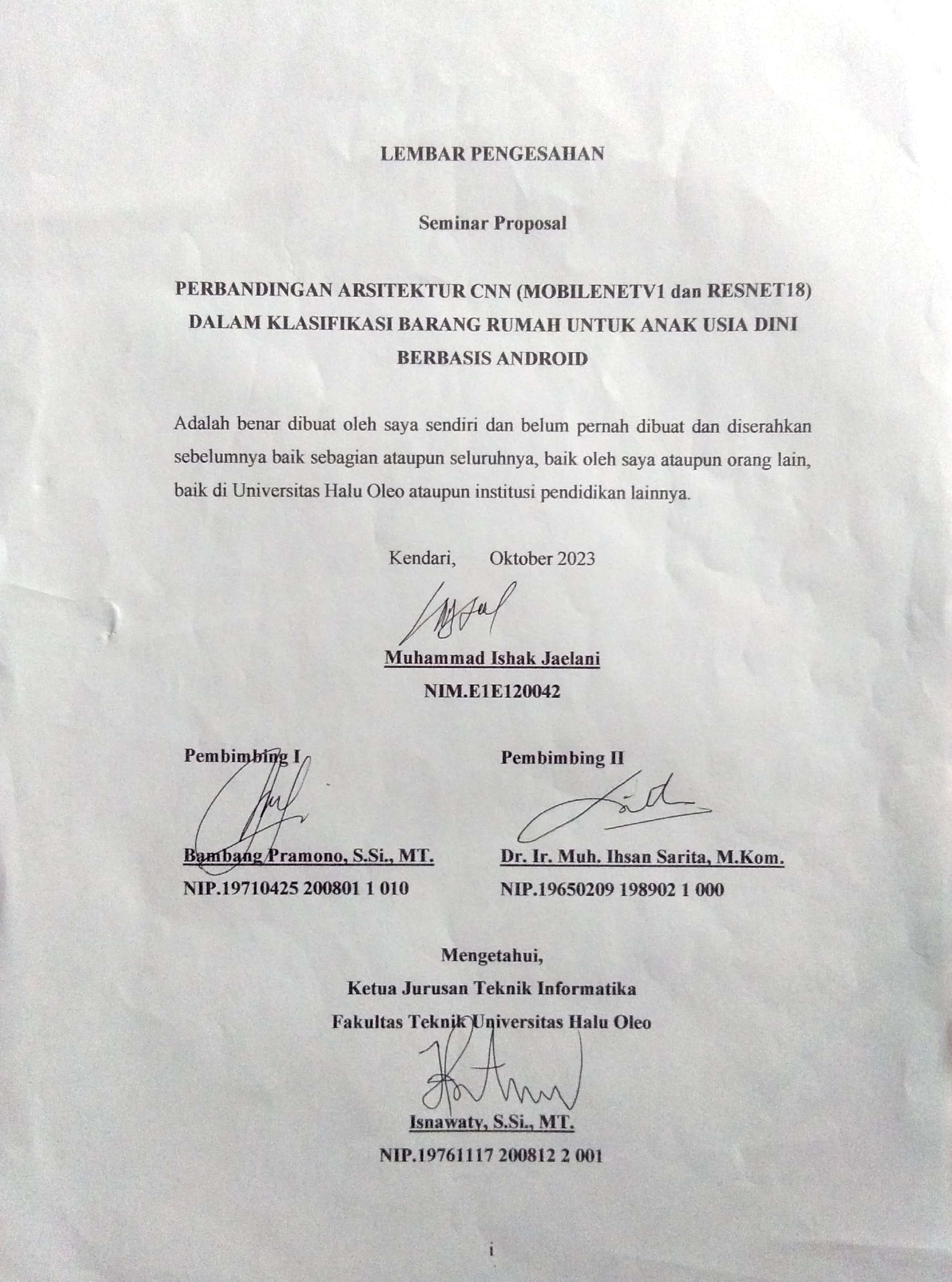
**FAKULTAS TEKNIK**

**UNIVERSITAS HALU OLEO**

**KENDARI**

**2023**

# LEMBAR PENGESAHAN



# INTISARI

**Muhammad Ishak Jaelani**, E1E120042

**PERBANDINGAN ARSITEKTUR CNN (MOBILENETV1 DAN RESNET18) DALAM KLASIFIKASI BARANG RUMAH UNTUK ANAK USIA DINI BERBASIS ANDROID**

Proposal, Fakultas Teknik, 2023

**Kata Kunci**: *Convolutional Neural Network, Mobile Network V1, Residual Network 18, Flask, Android.*

Kesibukan di dalam pekerjaan dapat mengakibatkan banyak hal, termasuk kurangnya waktu untuk bersama dengan keluarga. Hal ini dapat mengakibatkan orang tua yang bekerja tidak sempat untuk mengajarkan hal-hal baru kepada anaknya. Pengajaran ini sangat penting khususnya kepada anak dengan usia di bawah 5 tahun yang masih menjalani masa emas untuk berkembang.

Pengajaran anak dapat dilakukan dengan menggunakan teknologi pengenalan gambar (*image recognition*). Dalam menerapkan teknologi pengenalan gambar diperlukan metode, seperti *Convolutional Neural Network* (CNN). Penerapan metode CNN yang digabungkan dengan *android* dapat membuat anak belajar sendiri dalam mengenal barang-barang baru yang berada di sekitarnya. Tujuan dari penelitian ini adalah untuk dapat merancang dan membangun aplikasi *android* yang dapat mengklasifikasi barang rumah. Selain itu, penelitian ini bertujuan untuk membandingkan arsitektur *Mobile Network V1* dan *Residual Network 18* dalam klasifikasi barang rumah. Sistem yang akan dibangun dalam penelitian ini berbasis *android* dan menggunakan *flask framework* dalam pembuatan API.

Penelitian ini akan menggunakan dua jenis pengujian, yaitu pengujian *black box* dan *confusion matrix*. Pengujian *black box* akan digunakan untuk menguji sistem perangkat lunak *android* dan pengujian *confusion matrix* akan digunakan untuk membandingkan kemampuan arsitektur *Mobile Network V1* dan *Residual Network 18*.

# ABSTRACT

**Muhammad Ishak Jaelani**, E1E120042

***COMPARISON OF CNN ARCHITECTURES (MOBILENETV1 AND RESNET18) IN CLASSIFICATION OF HOME GOODS FOR EARLY CHILDREN BASED ON ANDROID***

*Proposal*, *Faculty of Engineering, 2023*.

***Keywords***: *Convolutional Neural Network, Mobile Network V1, Residual Network 18, Flask, Android.*

*Being busy at work can result in many things, including lack of time to spend with family. This can result in working parents not having time to teach their children new things. This teaching is very important, especially for children under 5 years of age who are still undergoing a golden period of development.*

*Teaching children can be done using image recognition technology. In applying image recognition technology, methods are needed, such as Convolutional Neural Network (CNN). The application of the CNN method combined with Android can make children learn on their own to recognize new items around them. The aim of this research is to be able to design and build an Android application that can classify home goods. In addition, this study aims to compare the Mobile Network V1 and Residual Network 18 architectures in home goods classification. The system that will be built in this research is Android-based and uses the Flask framework in creating APIs.*

*This research will use two types of testing, namely black box testing and confusion matrix. Black box testing will be used to test the Android software system and confusion matrix testing will be used to compare the capabilities of the Mobile Network V1 and Residual Network 18 architectures.*

# DAFTAR ISI

[LEMBAR PENGESAHAN i](#_Toc150690596)

[INTISARI ii](#_Toc150690597)

[ABSTRACT iii](#_Toc150690598)

[DAFTAR ISI i](#_Toc150690599)v

[DAFTAR TABEL vi](#_Toc150690600)

[DAFTAR GAMBAR vii](#_Toc150690601)

[BAB I](#_Toc150690602) [PENDAHULUAN 1](#_Toc150690603)

[1.1 Latar Belakang 1](#_Toc150690604)

[1.2 Rumusan Masalah 2](#_Toc150690605)

[1.3 Batasan Masalah 3](#_Toc150690606)

[1.4 Tujuan Penelitian 3](#_Toc150690607)

[1.5 Manfaat Penelitian 3](#_Toc150690608)

[1.6 Sistematika Penulisan **Error! Bookmark not defined.**](#_Toc150690609)

[1.7 Tinjauan Pustaka **Error! Bookmark not defined.**](#_Toc150690610)

[BAB II](#_Toc150690611) [LANDASAN TEORI 7](#_Toc150690612)

[2.1 Anak Usia Dini 7](#_Toc150690613)

[2.2 Barang Rumah 9](#_Toc150690614)

[2.2.1 Definisi 9](#_Toc150690615)

[2.2.2 Klasifikasi Barang Rumah 10](#_Toc150690616)

[2.2.3 Risiko Barang Rumah Bagi Anak Usia Dini 11](#_Toc150690617)

[2.3 Citra Digital 13](#_Toc150690618)

[2.4 Citra RGB 14](#_Toc150690619)

[2.5 *Preprocessing* Citra 15](#_Toc150690620)

[2.5.1 Augmentasi Data 15](#_Toc150690621)

[2.5.2 *Resize* 15](#_Toc150690622)

[2.5.3 Mengubah Citra RGB Menjadi Citra *Grayscale* 16](#_Toc150690623)

[2.6 *Convolutional Neural Network* 17](#_Toc150690624)

[2.7 *Residual Network18* (ResNet18) 20](#_Toc150690625)

[2.8 *Mobile Network V1* (MobileNetV1) 23](#_Toc150690626)

[2.9 HDF5 25](#_Toc150690627)

[2.10 *Android* 26](#_Toc150690628)

[2.11 *Flask* 26](#_Toc150690629)

[2.12 *Flowchart* 27](#_Toc150690630)

[2.13 *Unified Modeling Language* (UML) 28](#_Toc150690631)

[2.13.1 *Use Case* *Diagram* 28](#_Toc150690632)

[2.13.2 *Activity* *Diagram* 30](#_Toc150690633)

[2.13.3 *Sequence* *Diagram* 31](#_Toc150690634)

[2.13.4 *Class* *Diagram* 32](#_Toc150690635)

[2.14 Metodologi Pengembangan Sistem 32](#_Toc150690636)

[2.15 Pengujian *Black Box* 33](#_Toc150690637)

[2.16 Pengujian Akurasi *Confusion Matrix* 34](#_Toc150690638)

[BAB III](#_Toc150690639) [METODOLOGI PENELITIAN 36](#_Toc150690640)

[3.1 Metode Pengumpulan Data 36](#_Toc150690641)

[3.2 Metode Pengembangan Sistem 36](#_Toc150690642)

[3.3 Waktu dan Tempat Penelitian 37](#_Toc150690643)

[3.3.1 Waktu 37](#_Toc150690644)

[3.3.2 Tempat 37](#_Toc150690645)

[3.4 Analisis Kebutuhan Sistem 38](#_Toc150690646)

[3.4.1 Kebutuhan Fungsional 38](#_Toc150690647)

[3.4.2 Analisis Kebutuhan Nonfungsional 38](#_Toc150690648)

[3.5 Analisis Perancangan Sistem 40](#_Toc150690649)

[3.6 API *Request* dan *Response* 45](#_Toc150690650)

[3.7 *Preprocessing* 45](#_Toc150690651)

[3.7.1 Augmentasi Data 45](#_Toc150690652)

[3.7.2 *Resize* 46](#_Toc150690653)

[3.7.3 Konversi Citra RGB ke *Grayscale* 47](#_Toc150690654)

[3.8 Pengenalan Gambar 48](#_Toc150690655)

[3.8.1 *Residual* *Network 18* 49](#_Toc150690656)

[3.8.2 *Mobile Network V1* 52](#_Toc150690657)

[3.9 Perancangan UML 56](#_Toc150690658)

[3.10 Analisis Perancangan *Interface* 63](#_Toc150690659)

[3.11 Pengujian *Black Box* 67](#_Toc150690660)

[3.12 Pengujian *Confusion Matrix* 69](#_Toc150690661)

[DAFTAR PUSTAKA 71](#_Toc150690662)

# DAFTAR TABEL

[Tabel 2. 1 Standar Tingkat Pencapaian Perkembangan Anak 8](#_Toc150702824)

[Tabel 2. 2 Klasifikasi Barang Rumah 10](#_Toc150702825)

[Tabel 2. 3 Tabel Arsitektur ResNet18 20](#_Toc150702826)

[Tabel 2. 4 Tabel Arsitektur MobileNetV1 23](#_Toc150702827)

[Tabel 2. 5 Simbol-Simbol *Flowchart* 27](#_Toc150702828)

[Tabel 2. 6 Simbol-Simbol *Use Case Diagram* 29](#_Toc150702829)

[Tabel 2. 7 Simbol-Simbol *Activity Diagram* 30](#_Toc150702830)

[Tabel 2. 8 Simbol-Simbol *Sequence Diagram* 31](#_Toc150702831)

[Tabel 2. 9 Simbol-Simbol *Class Diagram* 32](#_Toc150702832)

[Tabel 2. 10 *Confusion Matrix* 34](#_Toc150702833)

[Tabel 3. 1 *Gantt Chart* Waktu Penelitian 37](#_Toc150702834)

[Tabel 3. 2 Kebutuhan Perangkat Keras 39](#_Toc150702835)

[Tabel 3. 3 Kebutuhan Perangkat Lunak 39](#_Toc150702836)

[Tabel 3. 4 Pengujian *Black Box System* 67](#_Toc150702837)

[Tabel 3. 5 Pengujian *Confusion Matrix* 69](#_Toc150702838)

# DAFTAR GAMBAR

[Gambar 2. 1 Anak Usia Dini 7](#_Toc150691013)

[Gambar 2. 2 Barang Rumah 12](#_Toc150691014)

[Gambar 2. 3 Citra Digital 14](#_Toc150691015)

[Gambar 2. 4 *Lux Light Pro Meter* 14](#_Toc150691016)

[Gambar 2. 5 RGB 15](#_Toc150691017)

[Gambar 2. 6 *Convolution Layer* 18](#_Toc150691018)

[Gambar 2. 7 *Max Pool* 19](#_Toc150691019)

[Gambar 2. 8 *Skip Connection* 22](#_Toc150691020)

[Gambar 2. 9 *Depthwise Separable Convolution* 25](#_Toc150691021)

[Gambar 2. 10 *Android* 26](#_Toc150691022)

[Gambar 2. 11 *Flask* 27](#_Toc150691023)

[Gambar 3. 1 Gambaran Keseluruhan Sistem Dari Sisi Pengguna 41](#_Toc150691024)

[Gambar 3. 2 Gambaran Keseluruhan Sistem Dari Sisi Pengguna 2 43](#_Toc150691025)

[Gambar 3. 3 Gambaran *Training* Model 44](#_Toc150691026)

[Gambar 3. 4 Gambaran *Training* Model 2 44](#_Toc150691027)

[Gambar 3. 5 API *Request* dan *Response* 45](#_Toc150691028)

[Gambar 3. 6 Augmentasi Data 46](#_Toc150691029)

[Gambar 3. 7 *Flowchart* *Resize* 47](#_Toc150691030)

[Gambar 3. 8 *Flowchart* Konversi Citra RGB ke *Grayscale* 48](#_Toc150691031)

[Gambar 3. 9 *Flowchart Residual Network* 18 50](#_Toc150691032)

[Gambar 3. 10 *Convolution* *Block* 51](#_Toc150691033)

[Gambar 3. 11 *Flowchart Residual Block* 52](#_Toc150691034)

[Gambar 3. 12 *Flowchart* MobileNetV1 Bagian 1 53](#_Toc150691035)

[Gambar 3. 13 *Flowchart* MobileNetV1 Bagian 2 54](#_Toc150691036)

[Gambar 3. 14 *Convolution Layer* dengan *Batch Normalization* dan ReLU 55](#_Toc150691037)

[Gambar 3. 15 *Depthwise Seperable Convolution* dengan *Batch Normalization* dan ReLU 55](#_Toc150691038)

[Gambar 3. 16 *Use Case Diagram* 56](#_Toc150691039)

[Gambar 3. 17 *Post Picture Activity Diagram* 57](#_Toc150691040)

[Gambar 3. 18 *About Activity Diagram* 57](#_Toc150691041)

[Gambar 3. 19 *Change Architecture Activity Diagram* 58](#_Toc150691042)

[Gambar 3. 20 *List of Items Activity Diagram* 58](#_Toc150691043)

[Gambar 3. 21 *Change the Item List View* *Activity Diagram* 59](#_Toc150691044)

[Gambar 3. 22 *Post Picture Sequence Diagram* 59](#_Toc150691045)

[Gambar 3. 23 *About Sequence Diagram* 60](#_Toc150691046)

[Gambar 3. 24 *About Sequence Diagram* 60](#_Toc150691047)

[Gambar 3. 25 *List of Items Sequence Diagram* 61](#_Toc150691048)

[Gambar 3. 26 *Change the Item List View Sequence Diagram* 61](#_Toc150691049)

[Gambar 3. 27 *Android Class Diagram* 62](#_Toc150691050)

[Gambar 3. 28 *API Class Diagram* 62](#_Toc150691051)

[Gambar 3. 29 Halaman *Splash Screen* 63](#_Toc150691052)

[Gambar 3. 30 Halaman *Dashboard* 64](#_Toc150691053)

[Gambar 3. 31 Menu 64](#_Toc150691054)

[Gambar 3. 32 Dialog Arsitektur 65](#_Toc150691055)

[Gambar 3. 33 Halaman Daftar Barang Mode 1 66](#_Toc150691056)

[Gambar 3. 34 Halaman Daftar Barang Mode 2 66](#_Toc150691057)

[Gambar 3. 35 Halaman *About* 67](#_Toc150691058)

# BAB I

# PENDAHULUAN

## Latar Belakang

Kesibukan di dalam pekerjaan adalah hal yang paling sering terjadi saat ini. Menurut Direktorat Statistik Kependudukan dan Ketenagakerjaan (2023), rata-rata jam kerja pekerja di Indonesia pada bulan februari tahun 2023 adalah 41 jam per minggu. Jam kerja ini masih belum ditambahkan dengan waktu lembur yang kadang didapatkan oleh para pekerja. Selain itu, waktu perjalanan dari tempat tinggal ke tempat kerja dapat menghabiskan sekitar 1 jam khususnya di kota besar seperti Jakarta. Kesibukan ini dapat memicu stres, tubuh mudah sakit dan kurangnya waktu dengan keluarga.

Kurangnya waktu dengan keluarga dapat menyebabkan beberapa kewajiban tidak dilaksanakan dengan baik khususnya untuk para orang tua yang bekerja. Salah satu pekerjaan rumah tangga untuk orang tua yang memiliki anak adalah mengajarkan hal-hal baru kepada anak. Anak memiliki rasa ingin tahu yang besar mengenai hal-hal yang baru dilihatnya khususnya untuk anak usia dini. Usia di bawah 5 tahun pada anak merupakan masa kritis dalam tumbuh kembang anak (waktu emas). Waktu emas ini merupakan waktu bagi anak untuk mengasah keterampilannya seperti pemahaman, komunikasi, motorik kasar, motorik halus dan lain-lain (Toghyani *et al.*, 2015).

Perkembangan teknologi yang cukup cepat dapat membantu permasalahan ini. Banyak aplikasi yang dirancang untuk dapat membantu dalam edukasi anak-anak. Contoh aplikasi tersebut adalah *Lingokids* dan *Intellecto Kids Learning Games*. Kedua aplikasi tersebut adalah aplikasi belajar digital yang dapat diakses kapan saja dan dimana saja. Selain itu, perkembangan teknologi ini telah menghasilkan sebuah teknologi pengenalan gambar (*Image Recognition*).

Pengenalan gambar adalah sebuah teknologi komputer yang digunakan untuk membuat komputer dapat mengenali gambar yang dimasukkan. Komputer akan memeriksa seluruh gambar dan mendapatkan karakteristik setiap gambar agar dapat mengenali objek, pola atau fitur di dalamnya. Teknologi pengenalan gambar

dapat digunakan sebagai alat membantu pembelajaran anak usia dini. Hal ini dikarenakan gambar merupakan alat visual yang efektif karena dapat divisualisasi. Selain itu, informasi yang disampaikan dari gambar dapat dimengerti dengan mudah karena hasil yang diragakan lebih mendekati kenyataan melalui foto yang diperlihatkan kepada anak-anak (Khotimah *et al.*, 2021). Salah satu algoritma untuk membuat teknologi pengenalan gambar adalah *Convolutional Neural Network* (CNN).

CNN adalah salah satu jenis algoritma yang mengajarkan komputer untuk memproses data dengan cara yang terinspirasi dari otak manusia. CNN menggunakan simpul atau *neuron* yang saling terhubung dalam struktur berlapis yang menyerupai otak manusia. CNN menciptakan sistem adaptif yang digunakan oleh komputer untuk belajar dari kesalahannya dan memperbaikinya secara terus-menerus.

CNN umumnya memiliki tiga tahapan, yaitu *convolution layer*, *pooling layer* dan *fully connected layer*. *Convolution layer* adalah tahapan dimana seluruh data akan dikonversi sehingga menghasilkan *activation map* atau *feature map* 2D. *Pooling layer* adalah lapisan yang berfungsi untuk mengurangi ukuran spasial dari fitur konvolusi sehingga dapat mengurangi sumber daya komputasi yang dibutuhkan untuk memproses data melalui pengurangan dimensi dari *feature map*. *Fully connected layer* adalah lapisan yang digunakan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear (P *et al.*, 2016). CNN memiliki beberapa arsitektur, seperti MobileNetV1 dan ResNet18. Perbedaan dari kedua arsitektur ini adalah susunan rancangan, seperti ukuran filter dan *stride* yang berbeda untuk setiap tahapan.

Oleh karena itu, penelitian yang akan dilakukan ini berjudul “**Perbandingan Arsitektur CNN (MobileNetV1 dan ResNet18) dalam Klasifikasi Barang Rumah untuk Anak Usia Dini Berbasis Android**”.

## Rumusan Masalah

Berdasarkan latar belakang permasalahan yang telah diuraikan di atas, dapat dirumuskan permasalahan antara lain:

1. Bagaimana merancang dan membangun aplikasi klasifikasi barang rumah berbasis *Android*.
2. Bagaimana perbandingan MobileNetV1 dan ResNet18 dalam klasifikasi barang rumah.

## Batasan Masalah

Adapun batasan masalah dalam penelitian ini yaitu:

1. Aplikasi ini dibuat untuk anak normal usia 1-4 tahun.
2. Data citra yang akan digunakan berjumlah 150 per kelas.
3. Barang rumah yang akan diklasifikasi dalam aplikasi ini adalah gelas mug kaca, pensil, gunting, bantal kepala, sikat gigi, sepatu, penghapus, buku, sendok, garpu dan sendal jepit.
4. Foto yang diambil hanya berisi 1 barang rumah.
5. Foto memiliki intensitas cahaya yang dapat membuat gambar barang rumah terlihat jelas.
6. Minimal versi *Android* yang digunakan adalah versi 6.0.1 (*Marshmallow*).

## Tujuan Penelitian

Adapun tujuan dalam penelitian ini, yaitu:

1. Merancang dan membangun sebuah aplikasi klasifikasi barang rumah berbasis *Android*.
2. Membandingkan kemampuan MobileNetV1 dan ResNet18 dalam klasifikasi barang rumah.

## Manfaat Penelitian

Adapun manfaat dalam penelitian ini, yaitu:

1. Penulis dapat merancang dan membangun sebuah aplikasi klasifikasi barang rumah berbasis *Android*.
2. Penulis dapat membandingkan kemampuan MobileNetV1 dan ResNet18 dalam klasifikasi barang rumah.

## 1.6 Sistematika Penulisan

**BAB I PENDAHULUAN**

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan tinjauan pustaka.

**BAB II LANDASAN TEORI**

Bab ini berisi pengertian-pengertian dan teori-teori yang menjadi acuan dalam pembuatan Analisa dan pemecahan dari permasalahan yang dibahas meliput *Convolutional Neural Network* (CNN), *Mobile Network* *V1* (MobileNetV1), *Residual Network 18* (ResNet18), *Android*, *Flask*, *Flowchart*, *Unified Modeling Language* (UML) dan pendukung lainnya.

**BAB III METODE PENELITIAN**

Bab ini memuat metodologi penelitian yang meliputi metode pengumpulan data, uraian metode *Rational Unified Process* (RUP) untuk pengembangan sistem dan uraian waktu penelitian.

**BAB IV ANALISIS DAN PERANCANGAN**

Bab ini memuat analisis serta rancangan sistem yang akan dibuat. Adapun rancangan sistem meliputi rancangan-rancangan proses, rancangan UML dan rancangan antarmuka sistem.

**BAB V IMPLEMENTASI DAN PENGUJIAN**

Bab ini menyajikan implementasi dan pengujian dari sistem yang dibangun. Bagian implementasi menguraikan tentang implementasi secara detail dan runut dari sistem yang dibangun berdasarkan hasil analisis dan rancangan pada bab sebelumnya sedangkan bagian pengujian menguraikan pengujian sistem serta pembahasan hasil pengujian sistem tersebut.

**BAB VI PENUTUP**

Bab ini memuat kesimpulan dari penelitian yang telah dilakukan serta saran untuk penelitian selanjutnya.

## 1.7 Tinjauan Pustaka

Beberapa penelitian yang dijadikan rujukan dalam penelitian ini yaitu Loekman & Lina, (2023) yang melakukan penelitian “Sistem Manajemen Inventori Dengan Pengenalan Barang Secara Otomatis Menggunakan Metode *Convolutional Neural Network*”. Hasil dari penelitian ini didapatkan bahwa *Convolutional Neural Network* sangat baik dalam hal pengenalan barang gudang dengan tingkat keberhasilan mencapai 93,94%. Selain itu, aplikasi dari penelitian ini adalah sebuah *website*. Penelitian ini menggunakan data training sebanyak 100 data untuk setiap class.

Penelitian kedua adalah penelitian yang dilakukan oleh Anhar & Putra, (2023) dengan judul penelitian “Perancangan dan Implementasi *Self-Checkout System* pada Toko Ritel menggunakan *Convolutional Neural Network* (CNN)”. Penelitian ini bertujuan untuk dapat mengklasifikasi tiga jenis produk, yaitu Teh Botol, Indomie dan Chitato. Penelitian ini menggunakan arsitektur MobileNetV2 dengan total jumlah data adalah 247. Aplikasi dari penelitian ini adalah aplikasi *desktop*. Hasil dari penelitian ini adalah algoritma CNN dengan arsitektur MobileNetV2 dapat menghasilkan akurasi 88.8%.

Penelitian ketiga adalah penelitian yang dilakukan Sandag & Waworundeng, (2021) dengan judul penelitian “Identifikasi Foto Fashion Dengan Menggunakan *Convolutional Neural Network* (CNN)”. Penelitian ini bertujuan untuk dapat mengklasifikasi beberapa jenis pakaian, seperti *T-Shirt*, *Trouser* dan *Pullover*. Penelitian ini menggunakan 60000 sampel *train* dan 10000 sampel *test* yang didistribusi secara merata dalam 10 kelas. Hasil dari penelitian ini adalah algoritma CNN dapat menghasilkan akurasi 93%.

Penelitian keempat adalah penelitian yang dilakukan Putri & Andjarwirawan, (2022) dengan judul penelitian “Penerapan Metode *Convolutional Neural Network* Untuk *Clothing Image Recognition*”. Penelitian ini bertujuan untuk dapat mengklasifikasi beberapa jenis pakaian, seperti *Kurtas*, *Rain Jacket* dan *Waistcoat*. Penelitian ini menggunakan arsitektur VGG16 dengan jumlah data per kelas adalah 1600 data *training* dan 400 data *testing*. Hasil dari penelitian ini adalah algoritma CNN dengan arsitektur VGG16 dapat menghasilkan akurasi sampai 94%.

Penelitian kelima adalah penelitian yang dilakukan Paul, (2022) dengan judul penelitian “*Classification of Garments from Fashion MNIST Dataset Using AlexNet CNN Architecture*”. Penelitian ini bertujuan untuk dapat mengklasifikasi beberapa jenis pakaian, seperti *T-Shirt*, *Trouser* dan *Pullover*. Penelitian ini menggunakan arsitektur AlexNet dengan jumlah data keseluruhan adalah 60000 sampel *train* dan 10000 sampel *test* yang didistribusi secara merata dalam 10 kelas. Hasil dari penelitian ini adalah algoritma CNN dengan arsitektur AlexNet dapat menghasilkan akurasi 91.78%.

Kesimpulan yang didapat dari kelima penelitian di atas adalah metode *Convolutional Neural Network* dapat digunakan sebagai metode pengenalan objek terutama barang dengan rata-rata tingkat keberhasilan yang baik. Sehingga penelitian ini mengusulkan untuk membandingkan dua arsitektur *Convolutional Neural Network* (MobileNetV1 dan ResNet18) dalam klasifikasi barang rumah untuk anak usia dini berbasis *Android*.

# BAB II

# LANDASAN TEORI

## Anak Usia Dini

Menurut Undang-Undang Nomor 23 Tahun 2002 tentang Perlindungan Anak, Bab I Pasal 1 ayat (1) menyatakan bahwa anak adalah orang yang belum berusia 18 tahun, termasuk yang masih dalam kandungan (Pemerintah Indonesia, 2002). Sedangkan menurut Undang-Undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional, Bab I Pasal 1 ayat (14), rentang usia dini adalah 0-6 tahun . Hal ini tergambar dalam pernyataan bahwa pendidikan anak usia dini adalah upaya pembinaan yang ditujukan kepada anak sejak lahir sampai dengan usia enam tahun melalui pemberian rangsangan pendidikan untuk membantu pertumbuhan dan perkembangan jasmani dan rohani agar anak memiliki kesiapan dalam memasuki pendidikan lebih lanjut (Pemerintah Indonesia, 2003).

Menurut Kementerian Pendidikan dan Kebudayaan Republik Indonesia (2014), anak usia dini adalah anak yang berada pada rentang usia 0-6 tahun, baik yang terlayani maupun yang tidak terlayani di Lembaga pendidikan anak usia dini. Menurut Suijono (2014), anak usia dini adalah anak yang baru dilahirkan hingga usia 6 tahun. Usia ini merupakan usia yang sangat menentukan bagi pembentukan karakter dan kepribadian anak serta kemampuan intelektualnya.



Gambar 2. 1 Anak Usia Dini

Toghyani et al. (2015) menyatakan bahwa usia di bawah 5 tahun pada anak merupakan masa kritis dalam tumbuh kembang anak (waktu emas). Waktu emas ini merupakan waktu bagi anak untuk mengasah keterampilannya seperti pemahaman, komunikasi, motorik kasar, motorik halus dan lain-lain. Standar isi tentang tingkat pencapaian perkembangan anak dapat dilihat pada tabel 2. 1 (Kementerian Pendidikan dan Kebudayaan Republik Indonesia, 2014).

Tabel 2. 1 Standar Tingkat Pencapaian Perkembangan Anak

|  |  |  |
| --- | --- | --- |
| **No.** | **Usia (Tahun)** | **Tingkat Pencapaian Perkembangan Anak** |
| 1. | 1 - 1,5 | 1. Menyebut beberapa nama benda, jenis makanan 2. Menanyakan nama benda yang belum dikenal 3. Membedakan ukuran benda (besar-kecil) 4. Menyebutkan bilangan tanpa menggunakan jari dari 1-10 tetapi masih suka ada yang terlewat 5. Menunjuk bagian tubuh yang ditanyakan 6. Menirukan bunyi, suara atau musik dengan irama yang teratur 7. Mencoret-coret |
| 2. | 1,5 - 2 | 1. Mempergunakan alat permainan dengan cara memainkannya tidak beraturan, seperti balok dipukul-pukul 2. Memahami gambar wajah orang 3. Merangkai puzzle 4. Memahami kata-kata sederhana dari ucapan yang didengar 5. Menyanyikan lagu sederhana 6. Anak mengenali musik dari program audio visual yang disukai |
| 3. | 2 - 3 | 1. Mulai meniru gerakan 2. Berjalan sambal berjinjit |

Tabel 2. 1 Standar Tingkat Pencapaian Perkembangan Anak (Lanjutan)

|  |  |  |
| --- | --- | --- |
| **No.** | **Usia (Tahun)** | **Tingkat Pencapaian Perkembangan Anak** |
| 3. | 2 - 3 | 1. Melempar dan menangkap bola 2. Melihat dan menyentuh benda yang ditunjukkan oleh orang lain 3. Meniru cara pemecahan orang dewasa atau teman 4. Mengenal bagian-bagian tubuh 5. Menggambar benda-benda lebih spesifik 6. Mengamati dan membedakan benda di sekitarnya yang di dalam rumah |
| 4. | 3 - 4 | 1. Paham bila ada bagian yang hilang dari suatu pola gambar seperti pada gambar wajah orang matanya tidak ada, mobil bannya copot dan lain-lain 2. Memahmi persamaan antara dua benda 3. Mengenal beberapa huruf atau abjad tertentu dari a-z yang pernah dilihatnya 4. Mulai mengikuti pola tepuk tangan 5. Mengenali alasan mengapa ada sesuatu yang tidak masuk dalam kelompok tertentu 6. Menggambar dengan menggunakan beragam media 7. Mengamati dan membedakan benda di sekitarnya yang di luar rumah |

## Barang Rumah

### Definisi

Barang adalah produk fisik yang berwujud, dapat dilihat dan disentuh yang digunakan masyarakat untuk memenuhi kebutuhannya atau untuk menghasilkan benda lain yang akan memenuhi kebutuhan masyarakat. Kebutuhan masyarakat dapat dibagi menjadi dua, yaitu kebutuhan primer dan kebutuhan sekunder. Kebutuhan primer adalah kebutuhan yang sangat penting untuk kelangsungan hidup manusia, seperti makanan, minuman, pakaian, dan tempat tinggal. Kebutuhan sekunder adalah kebutuhan yang tidak terlalu penting, tetapi dapat meningkatkan kualitas hidup manusia, seperti barang elektronik, kendaraan, dan hiburan (Manurung & Rahardja, 2008).

Menurut Undang-Undang Republik Indonesia Nomor 4 Tahun 1992 Tentang Perumahan dan Permukiman, Bab I Pasal 1 ayat (1), rumah adalah bangunan yang berfungsi sebagai tempat tinggal atau hunian dan sarana pembinaan keluarga (Pemerintah Indonesia, 1992). Dalam pengertian yang luas, rumah tinggal bukan hanya sebuah bangunan (struktural), melainkan juga tempat kediaman yang memenuhi syarat-syarat kehidupan yang layak, dipandang dari berbagai segi kehidupan masyarakat (Frick & Mulyani, 2006). Jadi, barang rumah adalah produk fisik yang berwujud, dapat dilihat dan disentuh yang digunakan di tempat tinggal atau hunian keluarga.

### Klasifikasi Barang Rumah

Klasifikasi barang rumah memiliki klasifikasi yang beragam, tergantung dari tujuan, fungsi, serta lokasi penempatannya. Klasifikasi barang rumah menurut ruangan penempatannya adalah sebagai berikut (Community Refugee Sponsors Australia, 2022).

Tabel 2. 2 Klasifikasi Barang Rumah

|  |  |  |
| --- | --- | --- |
| **No.** | **Ruangan** | **Nama Barang** |
| 1. | Dapur | Gelas minum, cangkir, piring, panci, gelas ukur, wajan, sendok wajan, sendok sup, teko teh, pisau, kompor, pembuka kaleng/botol, wadah penyimpanan makanan dan talenan |
| 2. | Kamar Mandi | Sikat toilet, sabun mandi, sabun cuci muka, sampo, sikat gigi, deodoran, pasta gigi, pisau cukur sekali pakai, krim cukur, kertas toilet dan tisu wajah |

,

Tabel 2. 2 Klasifikasi Barang Rumah (Lanjutan)

|  |  |  |
| --- | --- | --- |
| **No.** | **Ruangan** | **Nama Barang** |
| 3. | Kamar Tidur | Gantungan baju, lemari pakaian, meja belajar, jam alarm, selimut, seprai, bantal kepala dan bantal tidur |
| 4. | Ruang Keluarga | Kursi, meja, televisi, lukisan dan foto keluarga |
| 5. | Ruang Makan | Meja makan, kursi makan, lemari penyimpanan, piring, gelas |

### Risiko Barang Rumah Bagi Anak Usia Dini

Orang dikatakan selamat jika terhindar atau bebas dari ancaman atas faktor-faktor berisiko bahaya, seperti anak yang terhindar dari sakit, terluka dan memar ketika menggunakan barang rumah. Banyak hal tidak terduga yang mungkin terjadi ketika menggunakan barang rumah, seperti tertelan, terkilir, tertusuk, tersedak, tergigit, terjatuh dan bahaya lainnya (Kementerian Pendidikan, 2021). Kecelakaan saat menggunakan barang rumah bisa terjadi, diantaranya disebabkan oleh:

1. Barang rumah luar dibangun di area yang tidak tepat, misalnya dekat tempat pembuangan sampah, di bawah pohon yang mudah tumbang atau dekat jalan.
2. Barang rumah luar dibangun di atas tempat yang keras, seperti semen, penuh bebatuan dan lain sebagainya.
3. Barang rumah tetap digunakan meskipun sudah rusak. Kerusakan ditandai dengan ada bagian yang berkarat, putus, retak dan sebagainya.
4. Barang rumah memiliki sudut-sudut tajam, seperti paku yang tidak tertancap utuh, mur yang longgar dan sebagainya.
5. Barang rumah dipenuhi daun, semut dan lain-lain sehingga anak dapat terserang penyakit kulit dan digigit serangga.
6. Barang rumah mengandung racun atau zat yang berbahaya bagi tubuh, seperti pengharum ruangan, deterjen pakaian, pembersih kamar mandi dan sebagainya.



Gambar 2. 2 Barang Rumah

Barang rumah yang digunakan oleh anak baik barang rumah luar dan dalam ruang dapat berisiko bagi keselamatannya saat dimainkan. Berikut adalah beberapa faktor risiko yang bisa terjadi saat penggunaan barang rumah dalam ruang (Kementerian Pendidikan, 2021).

1. Barang rumah yang keropos atau lapuk

Barang rumah terbuat dari kayu seiring dengan waktu bisa mengalami pelapukan, pengkroposan atau jamuran, terutama jika diletakkan pada tempat yang tidak sesuai dan lembab. Hal ini bisa menjadi bahan perhatian untuk pengelola agar memastikan barang rumah dalam kondisi yang layak untuk digunakan.

1. Kerusakan pada barang rumah dan bagian yang lancip

Jenis barang rumah baik yang terbuat dari kayu maupun plastik memungkinakn ada salah satu bagian yang rusak atau bagian yang lancip. Hal ini perlu diperhatikan agar ketika bermain peserta didik tidak terluka seperti tertusuk atau tergores.

1. Potongan barang rumah ukuran kecil

Beberapa jenis barang rumah memiliki bagian-bagian yang kecil seperti potongan puzzle, yang dikhawatirkan bisa tertelan oleh anak. Pengawasan orang tua sangat diperlukan ketika anak sedang bermain agar hal tersebut tidak terjadi.

1. Bagian yang terkelupas atau berserat tajam

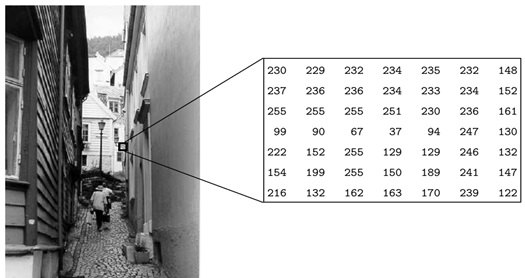
Jenis barang rumah berbahayan kayu perlu diperhatikan permukaannya apakah masih rata dan halus atau ada bagian yang terkelupas, sehingga menimbulkan serat kayu yang kasar. Serat kayu yang menonjol dikhawatirkan dapat melukai jari-jemari anak ketika bermain.

1. Barang rumah berbahan kain yang kotor dan berdebu

Barang rumah berbahan kain mudah sekali berdebu, kotor dan kusam. Barang rumah berbahan kain juga sangat mudah menyerap keringat dan menimbulkan bau, sehingga butuh dicuci secara berkala dan diberi kapur barus atau kamper di tempat penyimpanannya agar tidak menjadi media bersarangnya kuman dan bakteri serta menimbulkan alergi pada anak.

## Citra Digital

Citra adalah suatu gambar, foto ataupun berbagai tampilan dua dimensi yang menggambarkan suatu visualisasi objek. Citra dapat diwujudkan dalam bentuk tercetak ataupun digital. Citra digital adalah larik angka-angka secara dua dimensional. Citra digital tersimpan dalam suatu bentuk larik (*array*) angka digital yang merupakan hasil kuantifikasi dari tingkat kecerahan masing-masing piksel penyusun citra tersebut. Citra digital yang tersimpan dalam larik dua dimensi tersusun atas unsur-unsur kecil yang disebut dengan piksel. Masing-masing piksel terkait secara spasial dengan area di permukaan bumi. Struktur *array* ini tersusun dalam baris horizontal yang disebut baris (*lines*) dan kolom vertikal (*samples*). Masing-masing piksel dalam raster citra yang menyimpan nilai tingkat kecerahan piksel yang diwujudkan sebagai suatu angka digital. Susunan piksel dalam struktur *array* citra digital disebut data raster (Prabowo *et al.*, 2018).



Gambar 2. 3 Citra Digital

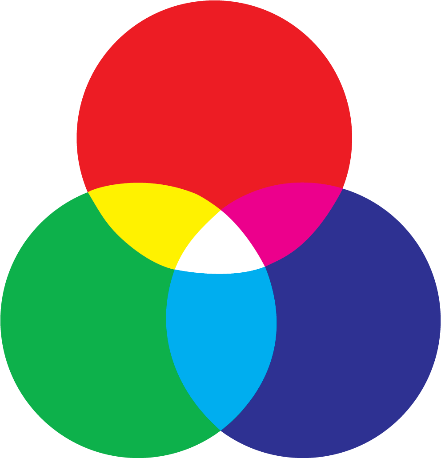
Pengambilan citra sangat dipengaruhi oleh intensitas cahaya. Cahaya yang sesuai diperlukan dalam pengambilan citra agar membuat objek terlihat jelas. Citra yang memiliki ukuran intensitas cahaya yang kurang akan menghasilkan citra yang lebih gelap, sedangkan citra yang memiliki ukuran intensitas cahaya yang berlebihan akan menghasilkan citra yang terlalu terang. Nurhidayat & Esye (2022) menjelaskan bahwa ukuran intensitas cahaya yang baik di dalam rumah adalah ±120 – 250 *lux* yang dapatdiukur dengan menggunakan aplikasi *Lux Light Meter Pro*.



Gambar 2. 4 *Lux Light Pro Meter*

## Citra RGB

Citra RGB adalah suatu citra yang memiliki model warna yang terdiri dari 3 buah warna, yaitu merah (*red*), hijau (*green*) dan biru (*blue*), yang ditambahkan dengan berbagai cara untuk menghasilkan bermacam-macam warna. Setiap warna mempunyai rentang 0-255, sehingga dapat menghasilkan 16.581.375 variasi warna berbeda pada gambar. Gambar tersebut bisa dikatakan sebagai gambar bit warna. Citra warna ini terdiri dari 3 *matrix* yang mewakili nilai dari *red*, *green* dan *blue* dalam setiap pixel (Fadjeri *et al.*, 2022).



Gambar 2. 5 RGB

## *Preprocessing* Citra

### Augmentasi Data

Augmentasi data merupakan proses mengubah atau memodifikasi gambar sedemikian rupa sehingga komputer akan mendeteksi bahwa gambar yang diubah adalah gambar yang berbeda, namun manusia masih dapat mengetahui bahwa gambar yang diubah tersebut adalah gambar yang sama. Augmentasi dapat meningkatkan akurasi dari model CNN yang dilatih karena dengan augmentasi model mendapatkan data-data tambahan yang dapat berguna untuk membuat model yang dapat melakukan generalisasi dengan baik. Contoh augmentasi data, yaitu membalikkan gambar secara horizontal dan melakukan rotasi gambar (Mahmud *et al.*, 2019).

### *Resize*

Tahapan ini terjadi proses normalisasi dimensi citra objek, yaitu proses pembesaran atau pengecilan dimensi citra objek menjadi dimensi yang telah ditentukan. Tujuannya, untuk menyamakan dimensi objek dari tiap citra yang dimasukkan, sehingga pada proses ekstraksi citra nanti tidak ada perbedaan dimensi dari matriks data citra objek. Ukuran baru hasil penskalaan didapat melalui perkalian antara ukuran citra input dengan variabel penskalaan. Berikut rumus persamaan yang digunakan dalam proses *scaling*.

|  |  |
| --- | --- |
|  | (2.1) |
|  | (2.2) |

Dengan (, ) adalah ukuran citra input, () adalah ukuran citra output dan (, ) adalah variabel penskalaan yang diinginkan. Jika variabel penskalaan bernilai lebih dari 1 maka hasil penskalaannya akan memperbesar ukuran citra, sebaliknya apabila variabel penskalaannya lebih kecil dari 1 maka hasilnya akan memperkecil ukuran citra. Salah satu metode untuk memperkecil ukuran citra adalah interpolation. Metode ini menggunakan nilai rata-rata suatu region untuk mewakili region tersebut (Norma, 2017). Salah satu algoritma *interpolation* adalah *bilinear interpolation*. *Bilinear interpolation* adalah algoritma yang melakukan penggandaan piksel berdasarkan piksel tetangga dekatnya, sehingga algoritma bilinear menghasilkan citra digital yang lebih baik dibandingkan yang digunakan untuk suatu piksel berdasarkan empat piksel. Persamaan menunjukkan cara menghitung nilai intesintas yang digunakan untuk suatu piksel berdasarkan empat piksel (Maulana, 2013).

|  |  |
| --- | --- |
|  | (2.3) |

### Mengubah Citra RGB Menjadi Citra *Grayscale*

Proses prapengolahan citra yang banyak dilakukan adalah mengubah citra warna (*true color*) menjadi citra keabuan (*grayscale*). Hal ini digunakan untuk menyederhanakan model citra. Perhitungan yang diterapkan pada citra warna dilakukan menggunakan tiga lapis (*layer*) yaitu R-*layer*, G-*layer* dan B-*layer*. Apabila menginginkan perhitungan yang sederhana maka citra warna dapat dikonversi menjadi citra keabuan yaitu mengubah 3 *layer* matriks citra warna menjadi 1 *layer* matriks citra keabuan (Achmad & Firadusy, 2013). Dalam proses mengubah citra berwarna menjadi citra *grayscale*, tiga layer RGB diubah menjadi 1 layer *grayscale*, tiga *layer* RGB diubah menjadi 1 *layer* matriks *grayscale*. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan.

Menurut Achmad (2013) citra warna dapat dikonversikan menjadi citra keabuan dengan operasi titik. Dalam definisi ringkasnya intesitas adalah nilai rerata dari ketiga elemen warna sehingga nilai keabuan yang mempresentasikan intesitas dapat dihitung dengan persamaan.

|  |  |
| --- | --- |
|  | (2.4) |

*Ko* sebagai nilai rerata dari ketiga komponen warna RGB yang diubah menjadi citra keabuan bukanlah suatu keharusan. Meskipun hasilnya cukup baik, penggunaan nilai rerata masih belum optimal untuk menunjukkan citra keabuan sehingga terkadang harus dilakukan pengubahan komposisi warna, sehingga persamaan dimodifikasi menajdi persamaan.

|  |  |
| --- | --- |
|  | (2.5) |

## *Convolutional Neural Network*

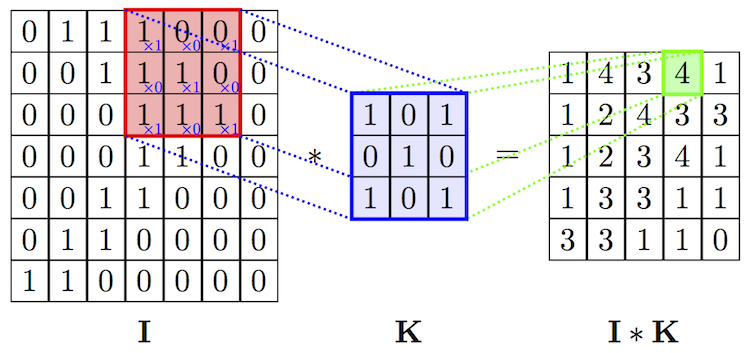
*Convolutional Neural Network* (CNN) adalah pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada kasus klasifikasi citra, MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan mengganggap setiap piksel adalah fitur yang independent sehingga menghasilkan hasil yang kurang baik (P, Wijaya & Soelaiman, 2016).

CNN pertama kali dikembangkan dengan nama NeoCognitron oleh Kunihiko Fukushima, seorang peneliti dari NHK Broadcasting Science Research Laboratories, Kinuta Setagaya, Tokyo, Jepang (Fukushima, 1980). Konsep tersebut kemudian dimatangkan oleh Yann LeChun, seorang peneliti dari AT&T Bell Laboratories di Holmdely, New Jersey, USA. Model CNN dengan nama LeNet berhasil diterapkan oleh LeChun pada penelitiannya mengenai pengenalan angka dan tulisan tangan (Cun *et al.*, 1990).

Sebuah CNN terdiri dari beberapa *layer*. Berdasarkan arsitektur pertama CNN, terdapat empat komponen utama pada sebuah CNN (P *et al.*, 2016), yaitu

1. *Convolution* *Layer*

*Convolution layer* melakukan operasi konvolusi pada *output* dari *layer* sebelumnya. *Layer* tersebut adalah proses utama yang mendasari sebuah CNN. Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada *output* fungsi lain secara berulang. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah *kernel* (kotak biru) pada citra di semua *offset* yang memungkinkan. Kotak paling kiri secara keseluruhan adalah citra yang akan dikonvolusi. *Kernel* bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar di sebelah kanannya. *Convolution layer* akan menghasilkan *feature maps* dari gambar juga mengandung *filter* (*convolutional filter*) yang mengkonversi gambar. Jumlah *feature maps* akan sama dengan jumlah *convolutional filter*.

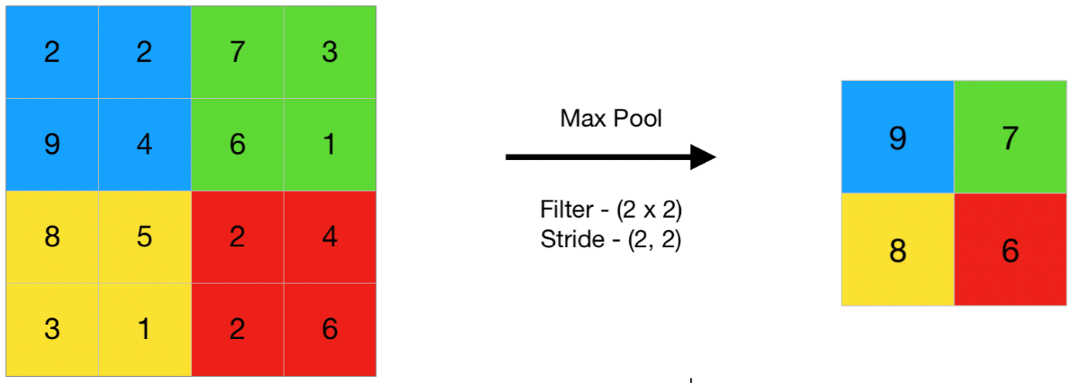


Gambar 2. 6 *Convolution Layer*

Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra *input*. Konvolusi akan menghasilkan transformasi *linear* dari data *input* sesuai informasi spasial pada data. Bobot pada *layer* tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga *kernel* konvolusi dapat dilatih berdasarkan input pada CNN.

1. *Pooling* *Layer*

*Pooling layer* adalah proses mereduksi ukuran sebuah data citra. Dalam pengolahan citra, *pooling* juga bertujuan untuk meningkatkan invariansi posisi dari fitur. Dalam sebagian besar CNN, metode pooling yang digunakan adalah *max* *pooling*. *Max* *pooling* membagi *output* dari *convolution* *layer* menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun matriks citra yang telah direduksi yang ditujukkan pada gambar . *Grid* yang bewarna merah, hijau, kuning dan biru merupakan kelompok *grid* yang akan dipilih nilai maksimumnya. Sehingga, hasil dari proses tersebut dapat dilihat pada kumpulan *grid* di sebelah kanannya. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun objek citra mengalam translasi (pergeseran).



Gambar 2. 7 Max Pool

1. *Fully* *Connected* *Layer*

*Fully connected layer* adalah *layer* yang biasanya digunakan dalam penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Setiap neuron pada *convolution* *layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah *fully connected layer*. Lapisan *fully connected* mengambil input dari lapisan *convolutinal* atau *pooling* akhir yang dalam bentuk seperangkat metrik (peta fitur) dan metrik tersebut diratakan untuk membuat vektor. Kemudian, vektor ini dimasukkan ke dalam lapisan *fully connected* untuk menghasilkan hasil akhir.

1. Fungsi Aktivasi

Fungsi aktivasi adalah fungsi *non linear* yang memungkinkan sebuah jaringan syaraf tiruan untuk dapat mentransformasi data *input* menjadi dimensi yang lebih tinggi sehinga dapat dilakukan hyperlane sederhana yang memungkinkan dilakukan klasifikasi. Dalam CNN terdapat fungsi aktivasi digunakan yaitu fungsi *Rectified Linear Unit* (ReLU) dan *Softmax*. Fungsi ReLU adalah fungsi yang menerapkan setiap nilai negatif akan diubah menjadi nol, sementara nilai positif tetap tidak berubah. Fungsi *softmax* adalah fungsi yang mengubah output numerik menjadi distribusi probabilitas ketika output terbesar memiliki probabilitas yang lebih tinggi.

|  |  |
| --- | --- |
|  | (2.6) |

## *Residual Network18* (ResNet18)

ResNet18 merupakan salah satu jenis arsitektur *Convolutional Neural Network* (CNN). ResNet18 memiliki jumlah kedalaman layer 18. Jumlah 18 dihitung dari jumlah *convolution layer* dengan *filter* 7x7 berjumlah 1 *layer*, 3x3 berjumlah 16 *layer* serta 1x1 *fully connected layer*. Pada setiap *convolution layer* terdapat *batch normalization* dan ReLU. Pada bagian akhir *layer* ekstrasi fitur terdapat *average pooling*. Sedangkan, *layer* klasifikasi memiliki *fully connected layer* dan *softmax* sebagai aktivasi (He *et al.*, 2015).

Tabel 2. 3 Tabel Arsitektur ResNet18

|  |  |  |
| --- | --- | --- |
| ***Layer Name*** | ***Output Size*** | ***18-Layer*** |
| *Convolution Layer 1* | 112 x 112 | 7 x 7, 64, *stride* 2 |
| *Block 1* | 56 x 56 | 3 x 3 *max pool*, *stride* 2 |
| 2 |

Tabel 2. 3 Tabel Arsitektur ResNet18 (Lanjutan)

|  |  |  |
| --- | --- | --- |
| ***Layer Name*** | ***Output Size*** | ***18-Layer*** |
| *Block 2* | 28 x 28 | 2 |
| *Block 3* | 14 x 14 | 2 |
| *Block 4* | 7 x 7 | 2 |
|  | 1 x 1 | *average pool*, *fully connected layer*, *softmax* |

Adapun komponen-komponen yang tidak termasuk component utama di dalam arsitektur pertama CNN, yaitu:

1. *Batch* *Normalization*

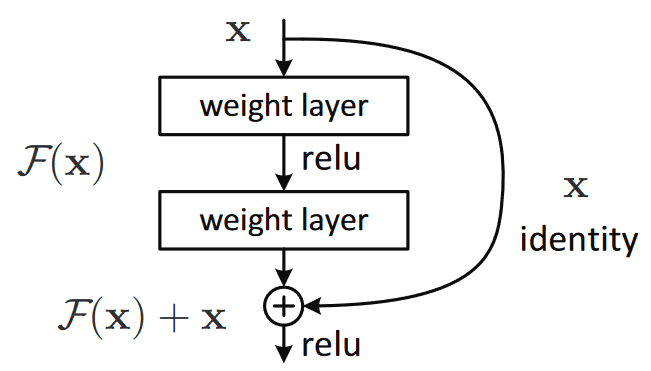
*Batch normalization* merupakan sebuah fungsi untuk menormalkan suatu nilai masukan yang masuk ke dalam fungsi aktivasi. *Batch normalization* merupakan sebuah metode terbaru yang digunakan dalam mengatasi masalah *gradient* yang hilang, yang dapat menyebabkan gradient aktivasi di lapisan yang berurutan berkurang atau bertambah besarnya. Selain itu, *batch* *normalization* digunakan untuk mengatasi masalah internal pergeseran kovariat. Prinsip kerja utama dari *batch* *normalization* adalah menskalakan nilai agar menjadi lebih kecil bertujuan untuk meringankan komputasi menjadi lebih singkat (Furusho & Ikeda, 2019).

1. *Block*

*Block* adalah komponen yang digunakan untuk memperoleh *feature* yang akan di-*train* pada proses *training*. Pada arsitektur ResNet18, digunakan empat *block* yang dimana *output* suatu *block* akan menjadi input untuk *block* selanjutnya. Pada masing-masing *block*, *input* pada awalnya akan melalui *convolution* *layer* 2 dimensi. Tetapi, adanya distribusi data yang berbeda pada setiap *batch*, maka perlu dilakukan normalisasi. Tanpa adanya normalisasi, proses *training* akan menjadi kurang optimal karena model menjadi kurang stabil.

Setelah melalui *layer batch normalization*, terdapat *activation* *function* ReLU yang membiarkan nilai positif dan mengubah seluruh nilai negatif menjadi 0. ReLU dapat membuat *output* tidak lagi memiliki nilai negatif. Nilai yang telah melalui *activation* *function* ReLU tersebut akan melalui *convolution* *layer* 2 dimensi dan *batch* *normalization* layer lagi. Setelah itu, terdapat *max* *pooling* *layer* untuk mengurangi *feature* yang dihasilkan (Vania *et al.*, 2023).

*Output* dari seluruh rangkaian sebelumnya dijumlahkan dengan *output* dari *convolution* *layer* 2 dimensi pertama. Proses ini dinamakan *bypass* *connection* atau *skip* *connection*. Hal ini dilakukan untuk menghindari permasalahan *vanishing gradient* ataupun *exploding gradient*. Dengan demikian, *feature*-*feature* yang mungkin mulai menghilang ataupun terlalu menonjol dapat distabilkan kembali. *Feature* tersebut akan diteruskan ke block selanjutnya untuk diproses.



Gambar 2. 8 Skip Connection

*Residual Network 18* memiliki dua jenis *bypass* *connection*, yaitu *identify* *bypass* dan *downsampling bypass*. *Identify bypass* adalah *bypass* yang melewati dua lapisan konvolusional tanpa pemrosesan lebih lanjut sebelum fungsi penjumlahan. *Downsampling* *bypass* adalah *bypass* yang perlu melakukan operasi konvolusi dan *batch* *normalization* terlebih dahulu terhadap dimensi *output* sebelumnya. Hal ini dilakukan karena terjadi perbedaan ukuran dimensi ouptut sebelumnya dan ukuran dimensi *output* yang akan dijumlahkan (Cheah *et al.*, 2021). Pada umumnya, *block* yang menggunakan *identify bypass* disebut *identify block* dan *block* yang menggunakan *downsampling bypass* disebut dengan *convolution block* (Agrawal *et al.*, 2023).

## *Mobile Network V1* (MobileNetV1)

MobileNetV1 merupakan salah satu jenis arsitektur *Convolutional Neural Network* (CNN) yang dapat menghasilkan model yang memiliki ukuran kecil baik dari jumlah parameter maupun ukuran model yang dihasilkan. Para peneliti dari *Google* membuat arsitektur CNN yang dapat digunakan untuk kebutuhan *mobile*. MobileNetV1 adalah sebuah arsitektur model yang dikembangkan untuk pengembangan aplikasi pada perangkat mobile ataupun perangkat lain yang memiliki keterbatasan sumber daya perangkat keras dengan mengurangi ukuran dan kompleksitas model menggunakan *depthwise separable convolutions*. Selain itu, setiap operasi konvolusi di dalam MobielNetV1 akan diikuti dengan operasi *batch* *normalization* dan fungsi aktivasi ReLU (Howard *et al.*, 2017). Perbedaan mendasar antara arsitektur MobileNet dan arsitektur CNN pada umumnya adalah penggunaan lapisan atau *layer* konvolusi dengan ketebalan *filter* yang sesuai dengan ketebalan *input image*.

Tabel 2. 4 Tabel Arsitektur MobileNetV1

|  |  |  |
| --- | --- | --- |
| ***Type / Stride*** | ***Filter Shape*** | ***Input Size*** |
| Conv / s2 | 3 x 3 x 32 | 224 x 224 x 3 |
| Conv dw / s1 | 3 x 3 x 32 dw | 112 x 112 x 32 |
| Conv / s1 | 1 x 1 x 32 x 64 | 112 x 112 x 32 |
| Conv dw / s2 | 3 x 3 x 64 dw | 112 x 112 x 64 |
| Conv / s1 | 1 x 1 x 64 x 128 | 56 x 56 x 64 |
| Conv dw / s1 | 3 x 3 x 128 dw | 56 x 56 x 128 |
| Conv / s1 | 1 x 1 x 128 x 128 | 56 x 56 x 128 |
| Conv dw / s2 | 3 x 3 x 128 dw | 56 x 56 x 128 |
| Conv / s1 | 1 x 1 x 128 x 256 | 28 x 28 x 128 |
| Conv dw / s1 | 3 x 3 x 128 dw | 28 x 28 x 256 |
| Conv / s1 | 1 x 1 x 256 x 256 | 28 x 28 x 256 |
| Conv dw / s2 | 3 x 3 x 256 dw | 28 x 28 x 256 |
| Conv / s1 | 1 x 1 x 256 x 512 | 14 x 14 x 256 |

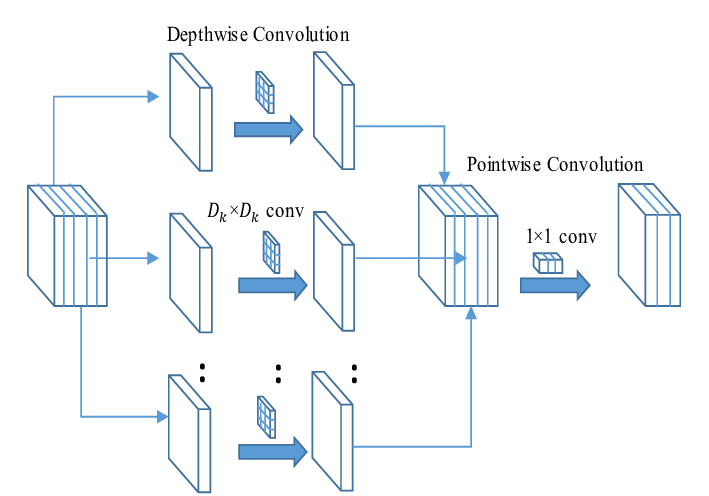
Tabel 2. 4 Tabel Arsitektur MobileNetV1 (Lanjutan)

|  |  |  |
| --- | --- | --- |
| ***Type / Stride*** | ***Filter Shape*** | ***Input Size*** |
| |  |  | | --- | --- | | 5 x | Conv dw /s1 | | Conv / s1 | | 3 x 3 x 512 dw  1 x 1 x 512 x 512 | 14 x 14 x 512  14 x 14 x 512 |
| Conv dw / s2 | 3 x 3 x 512 dw | 14 x 14 x 512 |
| Conv / s1 | 1 x 1 x 512 x 1024 | 7 x 7 x 512 |
| Conv dw / s2 | 3 x 3 x 1024 dw | 7 x 7 x 1024 |
| Conv / s1 | 1 x 1 x 1024 x 1024 | 7 x 7 x 1024 |
| *Average Pool* / s1 | *Pool* 7 x 7 | 7 x 7 x 1024 |
| *Fully Connected Layer* / s1 | 1024 x 1000 | 1 x 1 x 1024 |
| *Softmax* / s1 | *Classifier* | 1 x 1 x 1000 |

Adapun komponen yang tidak termasuk component utama di dalam arsitektur pertama CNN, yaitu:

1. *Depthwise* *Separable* *Convolution*

*Depthwise separable convolution* merupakan konvolusi faktorisasi yang memfaktorkan konvolusi standar menjadi depthwise *convolution* dan sebuah konvolusi 1 x 1 yang disebut *pointwise convolution*. *Depthwise convolution* membagi *input* dan *kernel*/*filter* konvolusinya ke dalam *channel depth* yang terpisah dan kemudian melakukan konvolusi pada setiap *channel input* dengan *channel filter* konvolusi yang sesuai. Setelah dihasilkan output dari masing-masing *channel*, semua *output* *channel* tersebut kemudian ditumpuk menjadi satu. *Pointwise convolution* melakukan konvolusi pada *output* proses depthwise *convolution* dengan filter berukuran 1 x 1 piksel untuk menggabungkan tumpukan *output channel* tersebut menjadi satu buah *channel* saja. Proses *depthwise separable* *convolution* menghasilkan output yang sama dengan proses konvolusi standar, namun proses ini melibatkan jumlah parameter yang lebih sedikit sehingga lebih efisien (Howard *et al.*, 2017).



Gambar 2. 9 Depthwise Separable Convolution

## HDF5

*Hierarchical Data Format version 5* (HDF5) adalah format penyimpanan data yang digunakan untuk mengatur, menyimpan, dan berbagi data secara efisien, terutama dalam konteks ilmu komputer, ilmu data, dan disiplin ilmu lainnya yang memerlukan pengelolaan data yang rumit. HDF5 memiliki beberapa karakteristik utama, yaitu (Yang *et al.*, 2005):

1. Penyimpanan Data Khusus

HDF5 menyediakan beberapa opsi penyimpanan data khusus untuk meningkatkan kinerja IO, efisiensi penyimpanan, dan manajemen data. Salah satu opsi penyimpanan data adalah chuncked. *Chuncked* adalah opsi penyimpanan yang digunakan untuk kumpulan data yang dapat diperluas dan dapat meningkatkan subsetting waktu akses.

1. Portabilitas

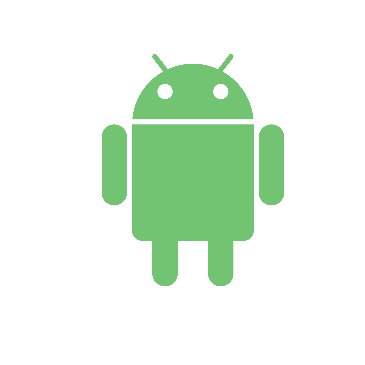
HDF5 mendukung *C*, *C++* dan *Fortran APIs*. HDF5 dapat berjalan di hampir semua komputasi ilmiah sistem, termasuk sistem paralel secara masif.

1. Paralel

Karakteristing penting lainnya adalah HDF5 mendukung IO paralel melalui MPI-IO. Hal ini memungkinkan aplikasi paralel untuk memanfaatkan sepenuhnya kekuatan lingkungan komputasi paralel.

## *Android*

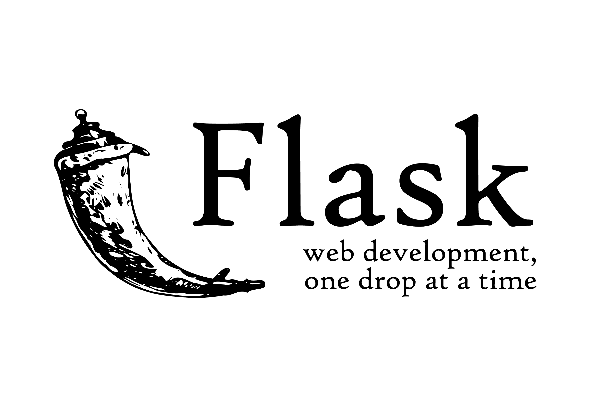
*Android* adalah sebuah sistem operasi perangkat *mobile* berbasi *Linux* yang mencangkup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri. *Android* juga menyediakan semua *tools* dan *framework* untuk mengembangkan aplikasi dengan mudah dan cepat. Dengan adanya *Android SDK* (*Software Development Kit*), pengembang aplikasi dapat membuat aplikasi pada platfrom *Android* menggunakan bahasa pemrograman *Java* atau *Kotlin* (Ahdan *et al.*, 2019).



Gambar 2. 10 Android

## *Flask*

*Flask* merupakan *web framework* yang ditulis dalam bahasa *Python*. *Flask* *framework* tergolong ke dalam *micro-framework* karena tidak membutuhkan *tools* atau *library* tertentu serta memiliki *database* bawaan. *Flask* merupakan framework dengan kinerja yang jauh lebih ringan dan cepat serta mudah diimplementasikan dibandingkan dengan *framework* *Python* lain. Hal ini dikarenakan *Flask* dibuat dengan tujuan untuk menyederhanakan inti *framework* seminimal mungkin. Pengembang atau *programmer* dapat membuat sebuah *website* ringan yang pastinya tetap terstruktur dan dapat mengatur perubahan suatu *website* serta *maintenance* dengan mudah karena sebagian besar fungsi dan komponen umum seperti *database*, validasi *form* dan sebagainya tidak terpasang secara *default* di *Flask* (Singh *et al.*, 2019)



Gambar 2. 11 Flask

Dalam penelitian ini, *Flask* akan digunakan sebagai *framework* dari *website* *Python* yang berisi *source* *code* dan model data klasifikasi barang rumah. Selain itu, *Flask* digunakan untuk membuat *Application Programming Interface* (API) yang akan menghubungkan *website* *Python* dan *Android*.

## *Flowchart*

Ada dua tool yang sering digunakan untuk membantu menyusun dokumen logika pemrograman, yaitu *flowchart* dan *pseudocode* (kode semu). *Flowchart* adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang sering terhubung. Jadi, setiap simbol *flowchart* melambangkan pekerjaan dan instruksinya. Simbol-simbol *flowchart* adalah standar yang ditentukan oleh *American National Standard Institute Inc* (eWolf Community, 2012).

Simbol-simbol yang ada pada *flowchart* diberikan pada Tabel 2. 5.

Tabel 2. 5 Simbol-Simbol Flowchart

|  |  |  |
| --- | --- | --- |
| **Nama Komponen** | **Deskripsi** | **Simbol** |
| *Start/End* | Simbol yang mendefinisikan awal atau akhir suatu *flowchart*. |  |
| *Process* | Simbol pemrosesan yang terjadi pada sebuah alur kerja. |  |
| *Input/Output* | Simbol yang mendefinisikan masukan dan keluaran proses. |  |

Tabel 2. 5 Simbol-Simbol *Flowchart* (Lanjutan)

|  |  |  |
| --- | --- | --- |
| **Nama Komponen** | **Deskripsi** | **Simbol** |
| *Decision* | Simbol untuk memutuskan proses lanjutan dari kondisi tertentu. |  |
| *On-Page Reference* | Simbol konektor untuk menyambung proses pada lembar kerja yang sama. |  |
| *Off-Page Reference* | Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda. |  |
| *Flowline* | Simbol untuk menghubungkan antar proses atau antar simbol. |  |
| *Database* | Simbol *database* atau basis data. |  |
| *Preparation* | Simbol yang digunakan untuk memberi nilai awal suatu variabel. |  |

## *Unified Modeling Language* (UML)

*Unified Modeling Language* (UML) merupakan kesatuan struktur dan cara bagi pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM (Pakaya *et al.*, 2020)

### *Use Case* *Diagram*

*Use Case diagram* atau *use case* adalah salah satu dari berbagai jenis diagram UML yang menggambarkan hubungan interaksi antara sistem dan *actor*. *Use case* dapat mendeskripsikan tipe interaksi antara si pengguna sistem dan sistemnya (Sukamto & Shalahudin, 2018).

Simbol-simbol yang ada pada *use case diagram* diberikan pada Tabel 2. 6.

Tabel 2. 6 Simbol-Simbol Use Case Diagram

|  |  |  |
| --- | --- | --- |
| **Nama Komponen** | **Deskripsi** | **Simbol** |
| *Use Case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau *actor* |  |
| *Actor* | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informaasi yang akan dibuat itu sendiri |  |
| *Association* | Komunikasi antara *actor* dan *use case* yang berpatisipasi pada *use case* atau *use case* memiliki hubungan interaksi dengan *actor* |  |
| *Generalization* | Menunjukkan spesialisasi *actor* untuk dapat berpatisipasi dengna *use case* |  |
| *Include* | Menunjukkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya |  |
| *Extend* | Menunjukkan bahwa suatu *use case* merupakan tambahan fungsional dari *use case* lainnya jika suatu kondisi terpenuhi |  |

### *Activity* *Diagram*

*Activity diagram* atau diagram aktivitas adalah diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem, Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity diagram* merupakan pengembangan dari *use case* yang memiliki alur aktivitas. Alur atau aktivitas berupa runtutan menu-menu atau proses bisnis yang terdapat di dalam sistem tersebut (Sukamto & Shalahudin, 2018).

Simbol-simbol yang ada pada *activity diagram* diberikan pada Tabel 2. 7.

Tabel 2. 7 Simbol-Simbol Activity Diagram

|  |  |  |
| --- | --- | --- |
| **Nama Komponen** | **Deskripsi** | **Simbol** |
| Status Awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal |  |
| Aktivitas | Aktivitas yang dilakukan sistem, aktivitasi biasanya diawali dengan kata kerja |  |
| *Decision* | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |  |
| *Join* | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan mnejadi satu |  |
| Status Akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir |  |
| *Swimlane* | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |  |

### *Sequence* *Diagram*

*Sequence diagram* atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu, *sequence* *diagram* juga akan menampilkan pesan atau perintah yang dikirm beserta waktu pelaksanaannya (Sukamto & Shalahudin, 2018).

Simbol-simbol yang ada pada *sequence diagram* diberikan pada Tabel 2. 8.

Tabel 2. 8 Simbol-Simbol Sequence Diagram

|  |  |  |
| --- | --- | --- |
| **Nama Komponen** | **Deskripsi** | **Simbol** |
| Aktor | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri |  |
| *Lifeline* | Menyatakan kehidupan suatu objek |  |
| Objek | Menyatakan objek yang berinteraksi pesan |  |
| Waktu Aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi |  |
| Pesan tipe *send* | Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya |  |
| Pesan tipe *return* | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi |  |
| Pesan tipe *self* | Menyatakan bahwa suatu objek mengirimkan pesan ke objek itu sendiri |  |

### *Class* *Diagram*

*Class diagram* atau diagram kelas adalah salah satu jenis diagram struktur pada UML yang menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, metode dan hubungan dari setiap objek (Sukamto & Shalahudin, 2018).

Simbol-simbol yang ada pada *class diagram* diberikan pada Tabel 2. 9.

Tabel 2. 9 Simbol-Simbol Class Diagram

|  |  |  |
| --- | --- | --- |
| **Nama Komponen** | **Deskripsi** | **Simbol** |
| Kelas | Kelas pada struktur sistem |  |
| Asosiasi/  *association* | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity* |  |
| Generalisasi | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus) |  |
| Kebergantungan/  *dependency* | Relasi antar kelas dengan makna kebergantungan antar kelas |  |
| Agregasi/  *aggregation* | Relasi antar kelas dengan makna semua-bagian (*whole-part*) |  |

## Metodologi Pengembangan Sistem

*Rational Unified Process* (RUP) merupakan salah satu metode pengembangan perangkat lunak dengan menggunakan pendekatan yang disiplin dalam melakukan setiap tugas dan tanggung jawabnya dalam sebuah organisasi. Tujuan dari RUP adalah untuk dapat menjamin produksi berkualitas tinggi dan memenuhi semua kebutuhan pihak yang berkepentingan, termasuk waktu dan biaya sesuai dengan rencana yang telah disepakati (Rachmat, 2022).

Adapun 4 tahapan kerja dari RUP, yaitu:

1. Fase *Inception* (Permulaan)

Fase *inception* adalah tahap pertama dalam pengembangan RUP. *Inception* dilakukan bertujuan untuk dilakukannya serangkaian analisis kebutuhan dalam sistem yang nantinya akan dikembangkan. Hasil dari fase ini pada pengembangan sistem yang dilakukan adalah pemodel proses bisnis *as-is* dan *to-be*, pemodelan *use case*, dan analisis persyaratan.x`

1. Fase *Elaboration* (Perluasan/Perencanaan)

Fase *elaboration* adalah fase melakukan perancangan atau arsitektur desain dari sistem yang akan dikembangkan. Hasil dari fase ini nantinya berupa *class diagram*, *sequence diagram*, perancangan antarmuka dan *physical data model*.

1. Fase *Construction* (Konstruksi)

Fase *construction* adalah fase melakukan pengimplementasian sistem berdasarkan perancangan yang sudah dilakukan sebelumnya dalam fase *elaboration* dan dilakukan pengujian terhadap sistem.

1. Fase *Transition* (Transisi)

Fase *transition* adalah fase melakukan *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Aktifitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

## Pengujian *Black Box*

Pengujian *black box* adalah pengujian aspek fundamental dari suatu sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan baik. Pengujian *black box* merupakan metode perancangan data pengujian berdasarkan spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi dalam perangkat lunak dan kemudian keluaran perangkat lunak diperiksa apakah sesuai dengan yang diharapkan. Pengujian *black box* mencoba untuk menemukan kesalahan dalam kategori (Susilawati *et al.*, 2022):

1. Fungsi yang salah atau hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja.
5. Kesalahan inisialisasi dan terminasi.

Pengujian *black box* memiliki beberapa tipe, salah satunya adalah tipe pengujian fungsional. Basis uji dari pengujian fungsional ini adalah pada spesifikasi dari komponen perangkat unak yang akan diuji. Pengujian fungsional berkosentrasi pada hasil dari proses, bukan bagaimana prosesnya terjadi (Anardani & Putera, 2019).

## Pengujian Akurasi *Confusion Matrix*

Pada bidang klasifikasi, ukuran akurasi dari suatu model klasifikasi merupakan hal yang penting untuk diperhatikan. Nilai akurasi dapat menggambarkan bagus tidaknya suatu model klasifikasi. *Confusion matrix* adalah sebuah matriks yang memuat data klasifikasi yang dilakukan oleh sistem klasifikas baik secara aktual maupun prediktif. Performa suatu model dapat diketahui dengan mengevaluasi data pada matriks. Pengolahan nilai-nilai yang ada pada kolom matriks *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), dan *True Positive* (TP) maka dapat diketahui nilai *accuracy*, *precision*, *recall* dan *F-measure* (Musfiroh *et al.*, 2021). Berikut tabel *confusion matrix* yang ditunjukkan pada table di bawah ini.

Tabel 2. 10 Confusion Matrix

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Prediksi | |
| Positif | Negatif |
| Aktual | Positif | TP | FN |
| Negatif | FP | TN |

Keterangan:

1. TP (*True Positive*) merupakan banyaknya data yang kelas aktualnya adalah kelas positif dengan kelas prediksinya merupakan kelas positif.
2. FN (*False Negative*) merupakan banyaknya data yang kelas aktualnya adalah kelas positif dengan kelas prediksinya merupakan kelas negatif.
3. FP (*False Positive*) merupakan banyaknya data yang kelas aktualnya adalah kelas negatif dengan kelas prediksinya merupakan kelas positif.
4. TN (*True Negative*) merupakan banyaknya data yang kelas aktualnya adalah kelas negatif dengan kelas prediksinya merupakan kelas negatif.

*Confusion matrix* dibagi dalam beberapa *performance matrix* popular yang paling sering digunakan, yaitu:

1. *Accuracy* (akurasi) menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. *Accuracy* merupakan rasion prediksi benar (positif dan negatif) dengan keseluruhan data. Dengan kata lain, *accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya).

|  |  |
| --- | --- |
|  | (2.7) |

1. *Precision* (presisi) menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Maka, *precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Dari semua kelas positif yang telah diprediksi dengan benar, berapa banyak data yang benar-benar positif.

|  |  |
| --- | --- |
|  | (2.8) |

1. *Recall* menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. Maka, *recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif.

|  |  |
| --- | --- |
|  | (2.9) |

# BAB III

# METODOLOGI PENELITIAN

## Metode Pengumpulan Data

Metode pengumpulan data yang digunakan untuk pengumpulan sejumlah data yang diperlukan dalam penyusunan tugas akhir ini, penulis menggunakan beberapa metode, yaitu:0

1. Kepustakaan, dilakukan dengan cara mengumpulkan referensi yang berkaitan dengan cara menggunakan dan membuat API. Selain itu, dilakukan juga pengumpulan referesnsi mengenai klasifikasi gambar menggunakan arsitektur MobileNetV1 dan ResNet18.
2. Studi literatur, dilakukan dengan cara membaca berbagai literatur, dokumen-dokumen serta arsip yang ada kaitannya dengan cara menggunakan dan membuat API. Selain itu, membaca berbagai literatur mengenai klasifikasi gambar menggunakan arsitektur MobileNetV1 dan ResNet18.
3. Untuk memenuhi kebutuhan data dalam penelitian ini, digunakan citra barang rumah dengan *training data* sebanyak 150 per kelas.

## Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam sistem ini adalah metode *Rational Unified Process* (RUP). Tahap ini memiliki empat fase dalam pengembangan sistem, yaitu:

1. *Inception* (Permulaan)

Pada fase ini dilakukan proses pengidentifikasian aplikasi, dilakukan dengan analisis kebutuhan akan aplikasi, melakukan kajian terhadap penelitian yang meliputi pengenalan *Flask*, algoritma *Convolutional Neural Network* (CNN), *Mobile NetworkV1* (MobileNetV1), *Residual Network18* (ResNet18).

1. *Elaboration* (Perluasan/perencanaan)

Pada tahap *elaboration* dilakukan proses analisis yaitu tahap ditentukan *Unified Modeling Language* (UML) yang terdiri dari *use case* *diagram*, *sequence* *diagram* dan *activity* *diagram* serta membuat *flowchart* dan analisis algoritma yang digunakan yaitu MobileNetV1 dan ResNet18. Pada tahap ini dilakukan juga desain arsitektur aplikasi.

1. *Construction* (Konstruksi)

Proses yang dilakukan dalam tahap ini ialah membangun perangkat lunak secara utuh, mulai dari interface aplikasi dan *coding* mulai dari *preprocessing*, *rescale*, *recognize* hingga penyambungan *website* dan *android* menggunakan API.

1. *Transition* (Transisi)

Fase *transition* difokuskan untuk melakukan proses *deployment*, untuk memastikan sistem sudah bekerja dengan baik di lingkungan pengguna. Dalam penelitian ini, sistem dipakai untuk melakukan pengujian terhadap aplikasi serta memperbaiki segala masalah yang muncul selama pengujian.

## Waktu dan Tempat Penelitian

### Waktu

Waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari bulan Oktober 2023 sampai dengan Januari 2024. Rincian kegiatan dapat dilihat pada Tabel 3.1 berikut:

Tabel 3. 1 Gantt Chart Waktu Penelitian

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Uraian | Waktu (2023-2024) | | | | | | | | | | | | | | | |
| Oktober | | | | November | | | | Desember | | | | Januari | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | *Inception* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | *Elaboration* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | *Construction* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | *Transition* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### Tempat

Penelitian tugas akhir ini dilakukan di Laborotarium *Software* *Engineering*, Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo.

## Analisis Kebutuhan Sistem

Analisis sistem adalah tahapan yang mempunyai tujuan untuk mengetahui dan mengamati semua yang terletak pada suatu sistem. Pembahasan yang ada pada analisis sistem ini yaitu analisis kebutuhan fungsional meliputi perancanga sistem menggunakan bahasa pemodelan UML, perancangan tampilan interface serta analisis kebutuhan nonfungsional yang meliputi kebutuhan perangkat keras dan perangkat lunak yang digunakan.

### Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah data yang dibutuhkan agar sistem dapat berjalan sesuai dengan prosedur. Setelah melalui tahap analisis kemudian akan ditetapkan kebutuhan-kebutuhan untuk membangun sistem. Pada analisis kebutuhan fungsional terdapat beberapa kebutuhan-kebutuhan dalam membangun sistem yaitu analisis kebutuhan input, analisis kebutuhan proses dan analisis kebutuhan output.

1. **Analisis Kebutuhan *Input***

*Input* dari aplikasi ini berupa data citra gambar barang rumah yang memiliki intensitas cahaya dan dapat membuat gambar terlihat jelas.

1. **Analisis Kebutuhan Proses**

Kebutuhan proses pada proses analisis sistem ini bertujuan mengidentifikasi permasalahan pada sistem, dimana aplikasi dibangun meliputi perangkat lunak dan analisis terhadap sistem.

1. **Analisis Kebutuhan *Output***

*Output* dari aplikasi ini adalah user dapat mengklasifikasi gambar barang rumah secara otomatis melalui *android smartphone*. Selain itu, penelitian ini dapat menghasilkan data perbandingan kemampuan arsitektur ResNet18 dan MobileNetV1 dalam klasifikasi barang rumah.

### Analisis Kebutuhan Nonfungsional

Analisis kebutuhan nonfungsional adalah sebuah langkah dimana pembangun aplikasi menganalisis sumber daya kebutuhan untuk membangun aplikasi yang akan dibangun. Analisis kebutuhan nonfungsional yang dilakukan dibagi dalam dua tahap, yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak. Kebutuhan perangkat keras yaitu kebutuhan perangkat atau komponen yang dibutuhkan pada sistem dan perangkat lunak yaitu kebutuhan perangkat lunak untuk membantu agar komponen perangkat keras dapat berfungsi dan dapat dijalankan pada sistem.

1. Kebutuhan perangkat keras (*Hardware*)

Untuk menerapkan rancangan yang telah dijelaskan sebelumnya, dibutuhkan beberapa perangkat keras sebagai sarana untuk mengimplementasikan aplikasi yang dibangun. Kebutuhan perangkat keras ditunjukkan pada Tabel 3. 2.

Tabel 3. 2 Kebutuhan Perangkat Keras

|  |  |  |
| --- | --- | --- |
| **No.** | **Nama Perangkat** | **Spesifikasi** |
| 1. | Laptop | Acer Swift SF314-511 |
|  | * *Processor* | 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz |
|  | * Monitor | 14 Inch |
|  | * Memori | Ram 16 GB, DDR5 Memory |
|  | * *Harddisk* | 500 GB SSD NVMe Kingston |
| 2. | *Android* *Smartphone* | Versi 6.0.1 (*Marshmallow*) |

1. Kebutuhan perangkat lunak (*Software*)

Adapun perangkat lunak yang dibutuhkan dalam penelitian ini ditunjukkan pada Tabel 3. 3.

Tabel 3. 3 Kebutuhan Perangkat Lunak

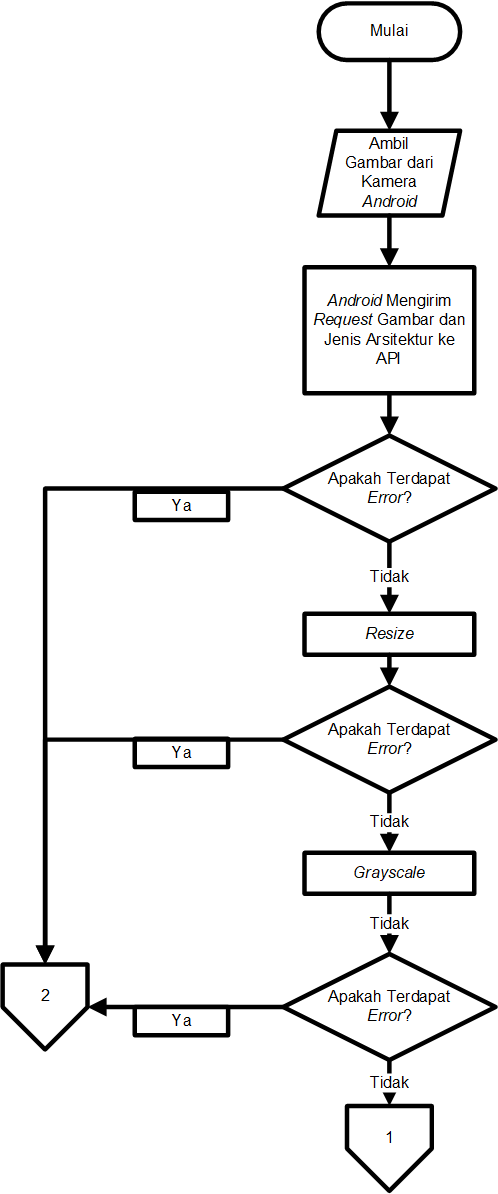
|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Perangkat** | **Fungsi** | **Spesifikasi** |
| 1. | *Windows* | *Operating System* | *Windows* 11 |
| 2. | *Android Studio* | *Integrated Development Environment* | Versi 2022.2 |

Tabel 3. 3 Kebutuhan Perangkat Lunak (Lanjutan)

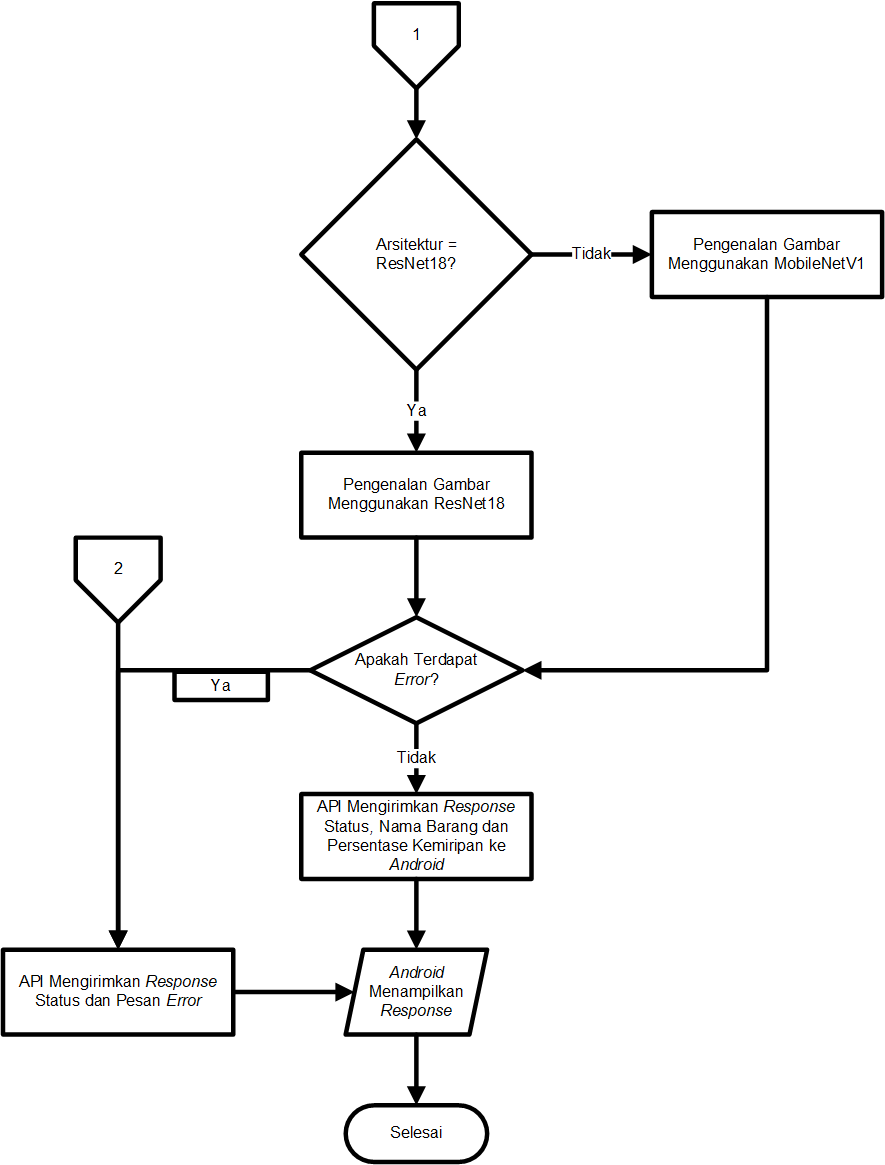
|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Perangkat** | **Fungsi** | **Spesifikasi** |
| 3. | *Visual Studio Code* | *Integrated Development Environment* | Versi 1.80.0 |
| 4. | *Chrome* | *Web Browser* | Versi 117.0.5938.132 |
| 5. | *Python* | Bahasa Pemrograman | Versi 3.10.4 |
| 6. | *Flask* | *Framework Python* | Versi 3.0.0 |
| 7. | *Postman* | *API Testing* | Versi 10.15.0 |
| 8. | *Tensorflow* | *Framework Python* | Versi 2.12.0 |
| 9. | *Kotlin* | Bahasa Pemrograman | Versi 1.8.20 |

## Analisis Perancangan Sistem

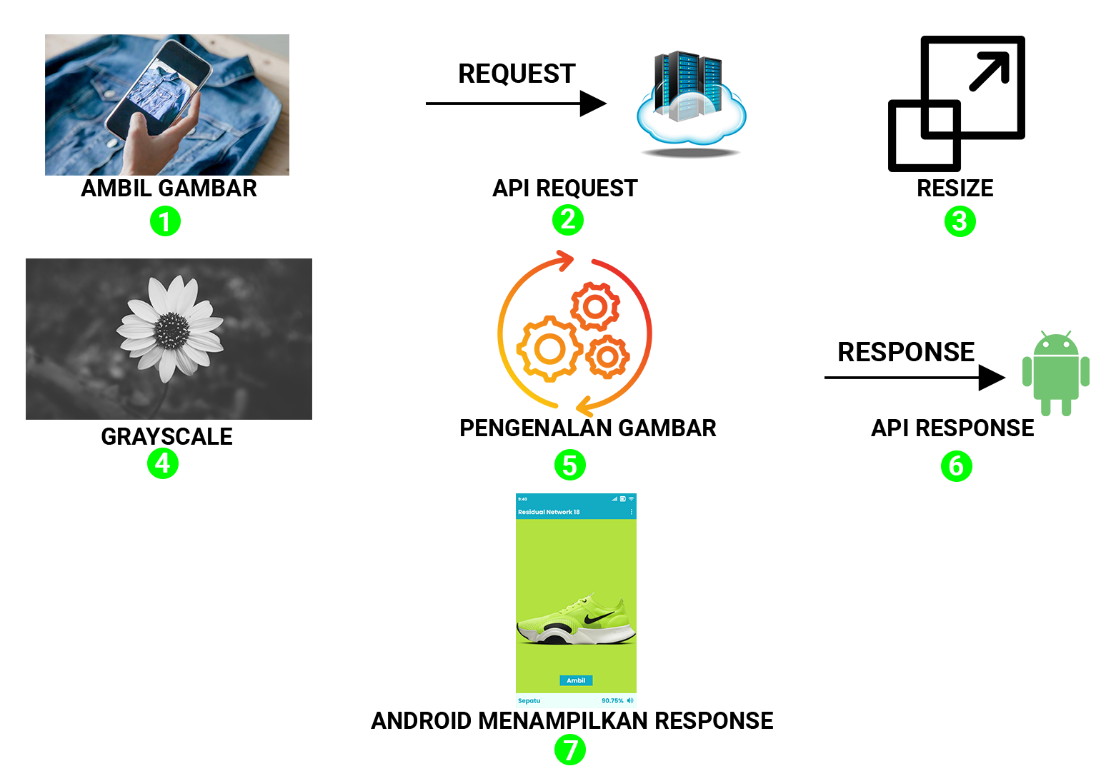
Sistem yang akan dibangun adalah sistem perbandingan klasifikasi CNN (MobileNetV1 dan ResNet18) dalam klasifikasi barang rumah untuk anak usia dini berbasis *android*. Sistem ini terdiri dari tiga proses utama yaitu API *request* dan *response*, *preprocessing* dan pengenalan gambar. Adapun perancangan penelitian yang diusulkan ditujukan pada gambar 3. 1, gambar 3. 2 dan gambar 3. 3.



Gambar 3. 1 Gambaran Keseluruhan Sistem Dari Sisi Pengguna



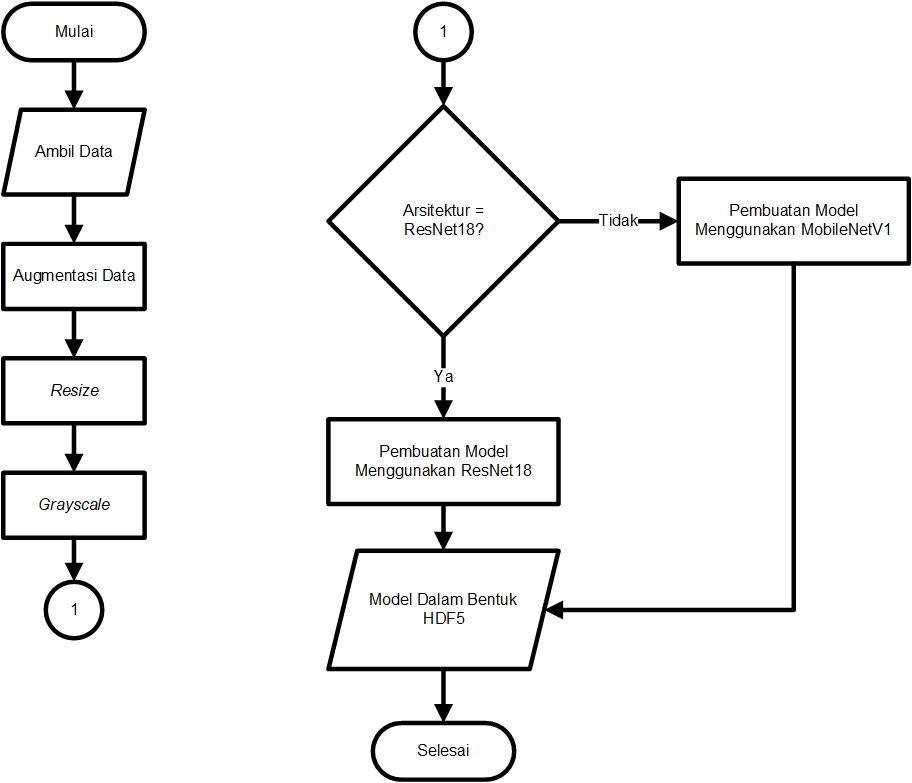
Gambar 3. 2 Gambaran Keseluruhan Sistem Dari Sisi Pengguna Bagian 2



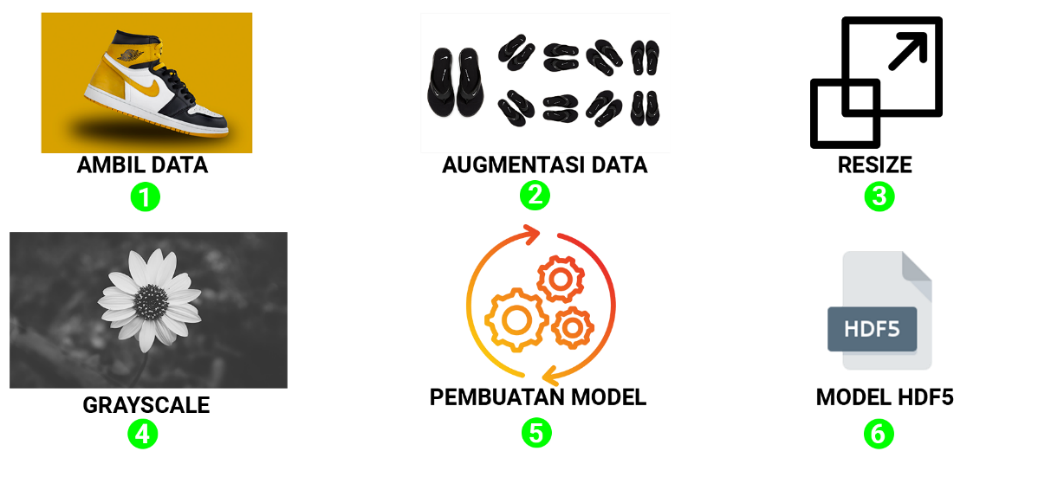
Gambar 3. 3 Gambaran Keseluruhan Sistem Dari Sisi Pengguna 2

Sistem yang akan dibangun memiliki beberapa tahapan. Tahapan pertama adalah ambil gambar. Pengambilan gambar barang dilakukan melalui kamera *android*. Pengguna harus mengarahkan kamera hanya ke satu barang saja. Tahapan kedua adalah API *request*. *Android* akan memngirimkan *request* berupa file foto dan jenis arsitektur (tipe data *string)* ke API. Tahapan ketiga adalah *resize*. API akan menyamakan ukuran setiap citra. Tahapan keempat adalah *grayscale*. API akan membuat citra rgb menjadi citra yang hanya memiliki warna tingkat keabuan. Tahapan kelima adalah pengenalan gambar. API akan melakukan proses pengenalan gambar sesuai dengna jenis arsitektur yang di-*request* oleh *android*. Tahapan keenam adalah API *response*. API akan memberikan *response* ke *android* berupa status (tipe data *string*), nama barang (tipe data *string*) dan persentase kemiripan (tipe data *float*) terhadap barang tersebut. Tahapan ketujuh adalah *android* akan menampilkan *response*. Jika, terdapat *error* pada salah satu tahapan tersebut, maka API akan memberikan *response status* (tipe data *string*) dan pesan *error* (tipe data *string*).

Tahapan *preprocessing* yang dilakukan di dalam gambaran keseluruhan sistem dari sisi pengguna sedikit berbeda dengan tahapan *preprocessing* yang dilakukan di dalam *training* model. Tahapan *preprocessing* di dalam *training* model terdapat satu tambahan tahapan, yaitu augmentasi data. Augmentasi data digunakan untuk memberikan variasi model yang lebih banyak. Selain itu, di dalam *training* model terdapat tahapan pembuatan model. Tahapan ini merupakan tahapan dimana model dibuat sesuai dengan arsitektur yang ditentukan.



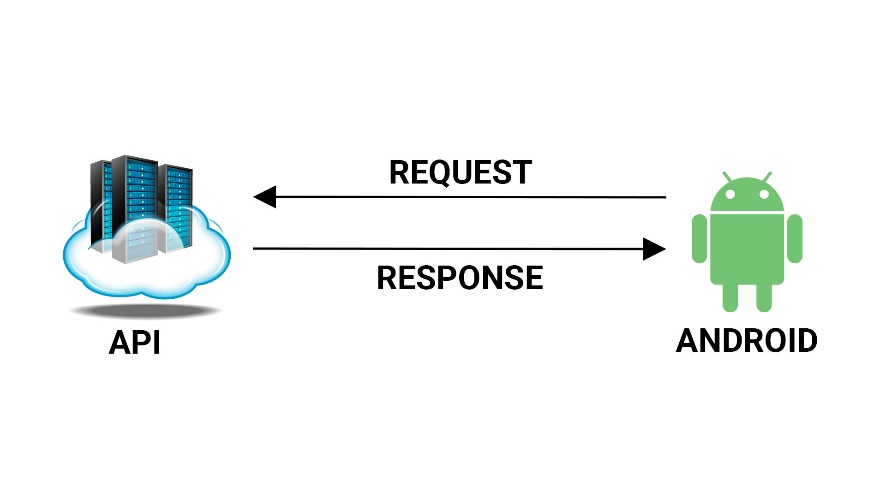
Gambar 3. 4 Gambaran *Training* Model



Gambar 3. 5 Gambaran *Training* Model 2

## API *Request* dan *Response*

Sistem ini akan dibangun menggunakan *Application Programming Interface* (API). API ini dibangun menggunakan bahasa *python* dan *framework* *flask*. Dalam sistem ini, *Android* akan memberikan *request* ke API berupa *file* gambar dan *text* jenis arsitektur yang dipilih. Lalu, sistem yang dibangun menggunakan bahasa *python* ini akan melakukan pengenalan gambar tersebut sesuai dengan jenis arsitektur yang diberikan. Setelah pengenalan gambar dilakukan, API akan memberikan *response* berupa nama barang yang bertipe data teks dan persentase kemiripannya yang bertipe data *float*.



Gambar 3. 6 API *Request* dan *Response*

## *Preprocessing*

### Augmentasi Data

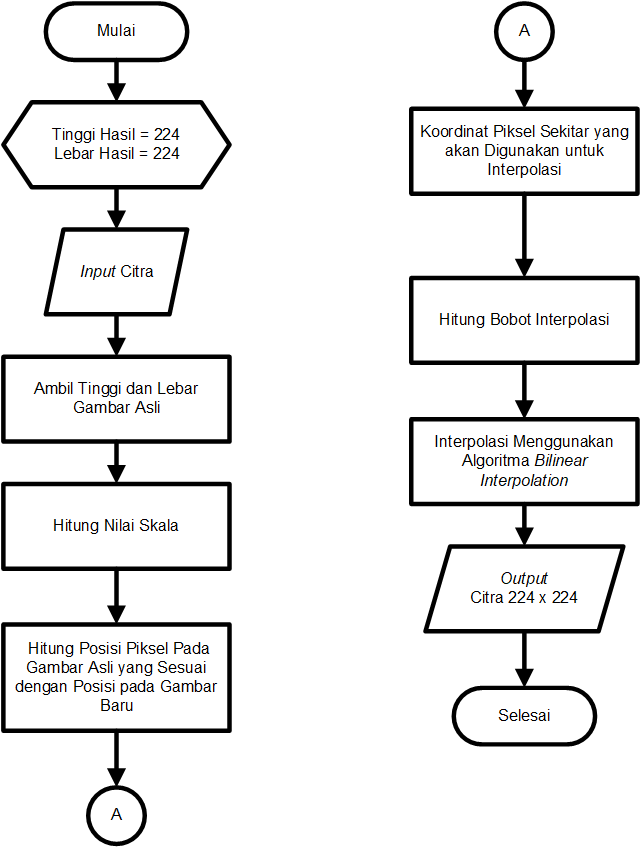
Augmentasi data yang dilakukan dalam penelitian ini adalah membalikkan gambar secara horizontal, rotasi searah jarum jam sebesar 45°, 90°, 135°, 180°, 225°, 270° dan 315°. Augmentasi data ini dilakukan agar dapat memberikan variasi gambar yang lebih banyak sehingga dapat memberikan model yang lebih baik.



Gambar 3. 7 Augmentasi Data

### *Resize*

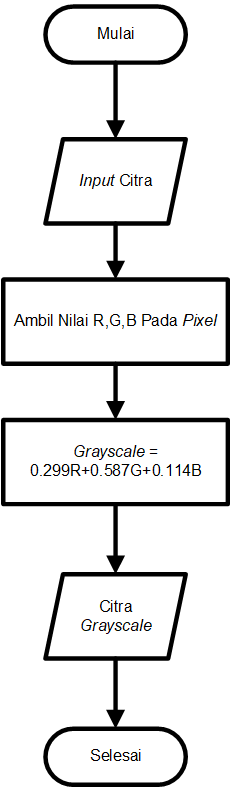
*Resize* yang dilakukan dalam penelitian ini akan menggunakan salah satu metode interpolasi yaitu algoritma *bilinear* *interpolation*. Dalam penelitian ini, *resize* berfungsi untuk sehingga model yang dibuat memiliki kemampuan yang lebih baik.



Gambar 3. 8 *Flowchart* *Resize*

### Konversi Citra RGB ke *Grayscale*

*Grayscale* adalah sebuah citra yang hanya memiliki warna tingkat keabuan. Tujuan konversi citra ini dilakukan agar dapat mempermudah model dalam mempelajari pola gambar yang terdapat di dalam data latih. Hal ini dikarenakan citra *grayscale* hanya memiliki satu *channel*, sedangkan citra RGB memiliki tiga *channel*.



Gambar 3. 9 *Flowchart* Konversi Citra RGB ke *Grayscale*

## Pengenalan Gambar

Pengenalan gambar adalah proses yang dijalankan oleh suatu sistem dalam mengenali gambar yang diberikan sesuai dengan model yang ada. Dalam penelitian ini, algoritma yang digunakan dalam pengenalan gambar adalah *Convolutional Neural Network* (CNN). Algoritma CNN memiliki 4 komponen utama, yaitu *convolution layer*, *pooling layer*, *fully connected layer* dan fungsi aktivasi.

* 1. *Convolution* *Layer*

*Convolution layer* bertujuan untuk mengekstraksi fitur dari citra. Citra *input* akan dikonversi menggunakan satu set kernel (juga disebut *filter*) yang dililitkan pada setiap bagian data citra dan menghasilkan citra baru yang mengalami peningkatan kompleksifitas yang disebut dengan *activation map* atau *feature map* 2D.

* 1. *Pooling* *Layer*

*Pooling* *layer* bertujuan untuk mereduksi ukuran sebuah data citra. Metode *pooling* *layer* yang digunakan dalam penelitian ini adalah *max pooling* dan *average* *pooling*. *Max* *pooling* adalah metode *pooling* yang membagi *output* dari *layer* sebelumnya menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun matriks citra yang telah direduksi. Sedangkan, *average* *pooling* adalah metode *pooling* yang mengambil nilai rerata dari setiap *grid* untuk menyusun matriks citranya.

* 1. *Fully* *Connected* *Layer*

*Fully connected layer* bertujuan untuk menggabungkan seluruh informasi *neurons* dari lapisan sebelumnya menajdi satu. Sebelum memasuki tahapan *fully connected layer*, *output* dari lapisan sebelumnya akan diubah menjadi data satu dimensi terlebih dahulu.

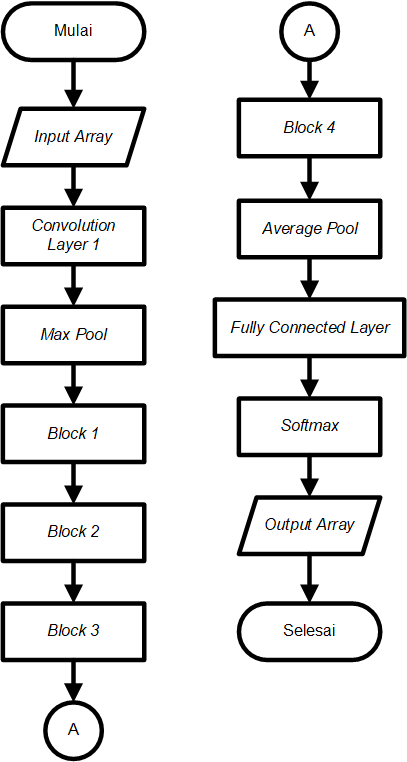
* 1. Fungsi Aktivasi

Fungsi aktivasi bertujuan untuk membuat *neuron* dapat memahami dan memproses data yang lebih kompleks dan non-linear. Hal ini dikarenakan fungsi aktivasi dapat membantu *neuron* mengenali pola-pola yang berbeda di dalam gambar-gambar. Fungsi aktivasi yang digunakan dalam penelitian ini, yaitu ReLU dan *softmax*.

Adapun arsitektur CNN yang akan dibandingkan dalam penelitian ini, yaitu arsitektur *Residual Network 18* (ResNet18) dan *Mobile Network V1* (MobileNetV1).

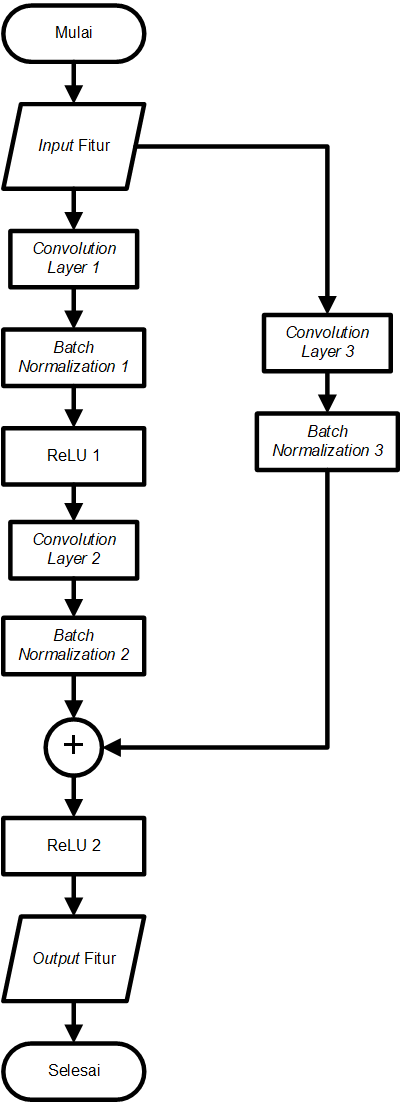
### *Residual* *Network 18*

*Residual Network 18* merupakan salah satu jenis arsitektur *Convolutional Neural Network* (CNN). Residual Network 18 memiliki jumlah kedalaman layer 18. Jumlah 18 dihitung dari jumlah *convolution layer* dengan *filter* 7 x 7 berjumlah 1 *layer*, 3 x 3 berjumlah 16 *layer* serta 1 x 1 *fully connected layer*. Adapun *flowchart residual network* *18* terdapat pada gambar 3. 10.



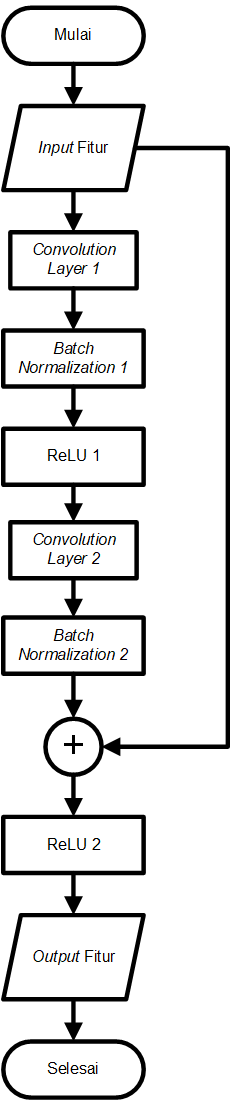
Gambar 3. 10 *Flowchart Residual Network* 18

Dalam penelitian ini, peneliti akan menggunakan dua jenis blok di dalam arsitektur ResNet18, yaitu *convolution* *block* dan residual *block*. *Convolution block* adalah blok yang memiliki downsampling *bypass* di dalamnya. *Convolution block* ini digunakan ketika terdapat perbedaan ukuran dimensi *output* fitur sebelumnya (*input* fitur untuk blok tersebut) dan ukuran dimensi *output* fitur yang akan dijumlahkan. Adapun *flowchart convolution block* terdapat pada gambar 3. 11.



Gambar 3. 11 *Convolution* *Block*

Sedangkan, *residual block* adalah blok yang memiliki *identify bypass* di dalamnya. Perbedaan *convolution block* dan *residual block* terletak pada keberadaan *convolution layer* dan *batch normalization* di dalam jalur alternatif (*bypass*). *Residual block* tidak memiliki dua proses tersebut di dalam *bypass* karena *residual block* digunakan ketika ukuran dimensi *output* fitur sebelumnya (*input* fitur untuk blok tersebut) dan ukuran dimensi *output* fitur yang akan dijumlahkan memiliki ukuran yang sama. Adapun *flowchart residual block* dapat dilihat pada gambar 3. 12.

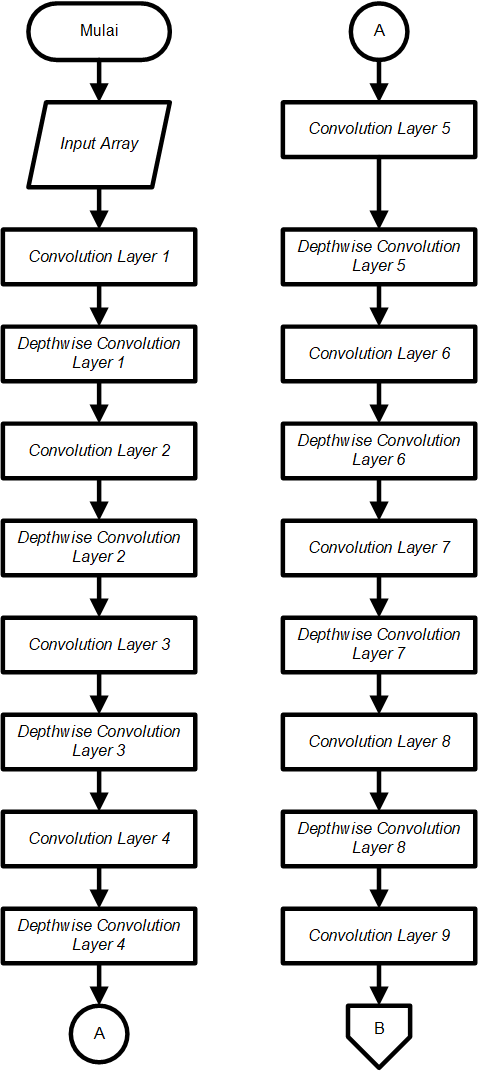


Gambar 3. 12 *Flowchart Residual Block*

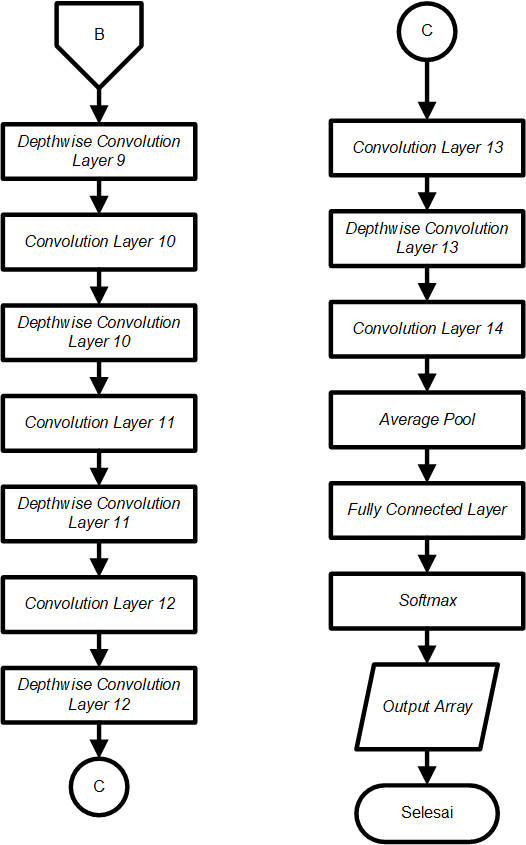
Selain *residual block* dan *convolution block*, arsitektur *residual network* 18 memiliki komponen *batch normalization*. *Batch normalization* digunakan untuk meringankan beban komputasi sehingga pembuatan model dapat menjadi lebih cepat.

### *Mobile Network V1*

*Mobile Network V1* (MobileNetV1) merupakan salah satu jenis arsitektur *Convolutional Neural Network* (CNN). MobileNetV1 memiliki beberapa komponen, yaitu 3 x 3 *convolution layer* berjumalah 1, 1 x 1 *convolution layer* berujumlah 13, 3 x 3 *depthwise separable convolution* berjumlah 13, 1 *average pool*,1 *fully connected layer* dan *1 softmax activation function*. Adapun *flowchart* MobileNetV1 dapat dilihat pada gambar 3. 13 dan 3. 14.

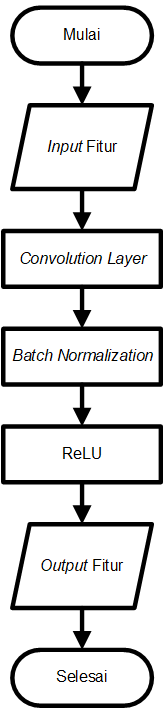


Gambar 3. 13 *Flowchart* MobileNetV1 Bagian 1

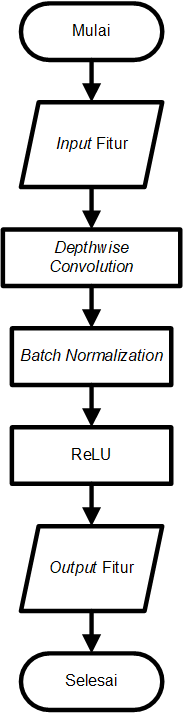


Gambar 3. 14 *Flowchart* MobileNetV1 Bagian 2

Dalam penelitian ini, setiap komponen *convolution layer* dan *depthwise separable convolution layer* akan diikuti dengan komponen *batch normalization* dan fungsi aktivasi ReLU. Sehingga, komponen MobileNetV1 ditambah 27 *batch normalization* dan 27 fungsi aktivasi ReLU.



Gambar 3. 15 *Convolution Layer* dengan *Batch Normalization* dan ReLU



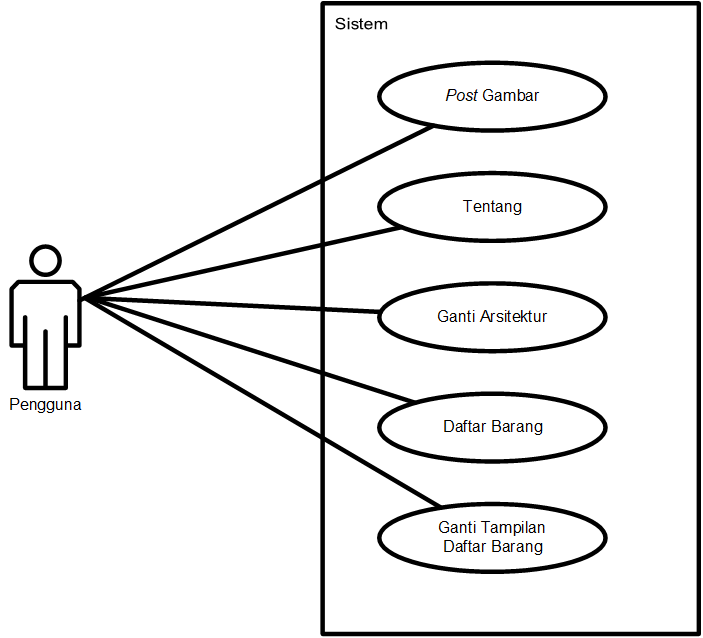
Gambar 3. 16 *Depthwise Seperable Convolution* dengan *Batch Normalization* dan ReLU

## Perancangan UML

Aplikasi dibangung dengan menggunakan *Unified Modelling Language* (UML). UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram yang terdiri dari *use case diagram*, *activity diagram*, *sequence diagram* dan *class diagram*.

1. *Use Case* *Diagram*

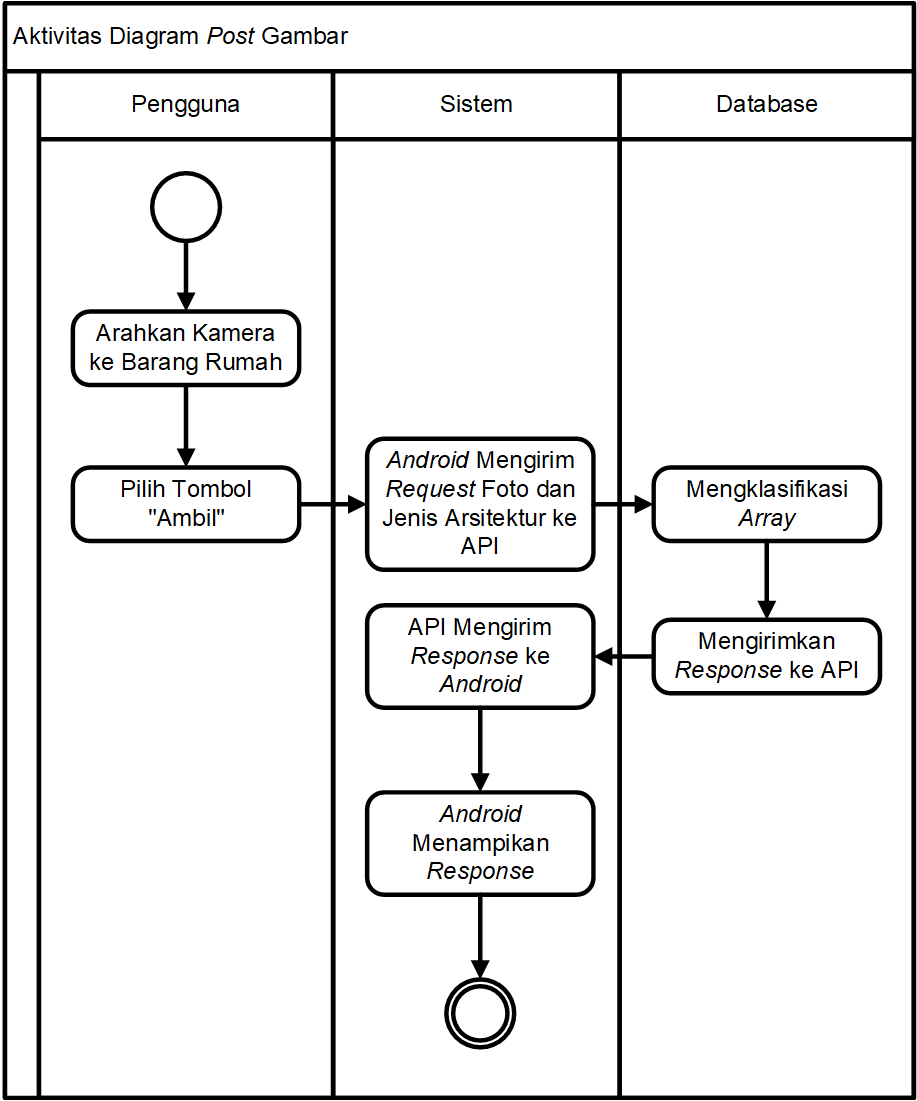
*Use case diagram* adalah seabuah diagram yang dapat merepresentasikan interaksi yang terjadi antara user dan sistem. *Use case diagram* ini mendeskripsikan bagaimana cara *user* berinteraksi dengan sistem. *Use case diagram* dari sistem yang akan dibangun adalah sebagai berikut:



Gambar 3. 17 *Use Case Diagram*

1. *Activity* *Diagram*
2. *Post Picture* *Activity Diagram*

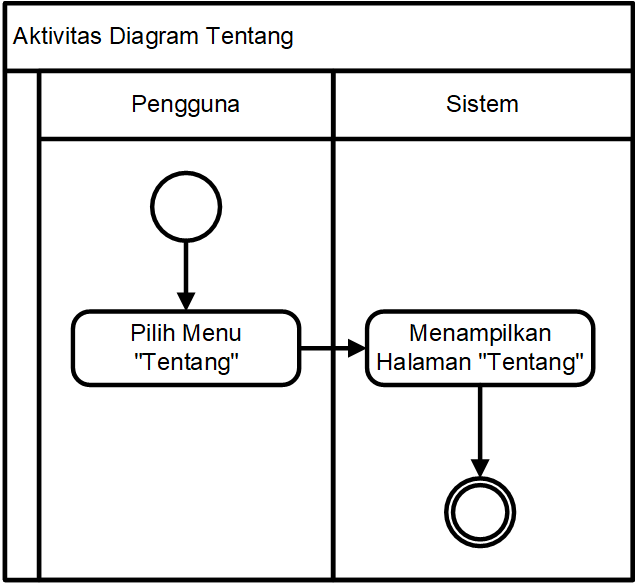
Adapun *post* *picture activity diagram* dapat dilihat pada gambar 3. 18.



Gambar 3. 18 *Post Picture Activity Diagram*

1. *About Activity* *Diagram*

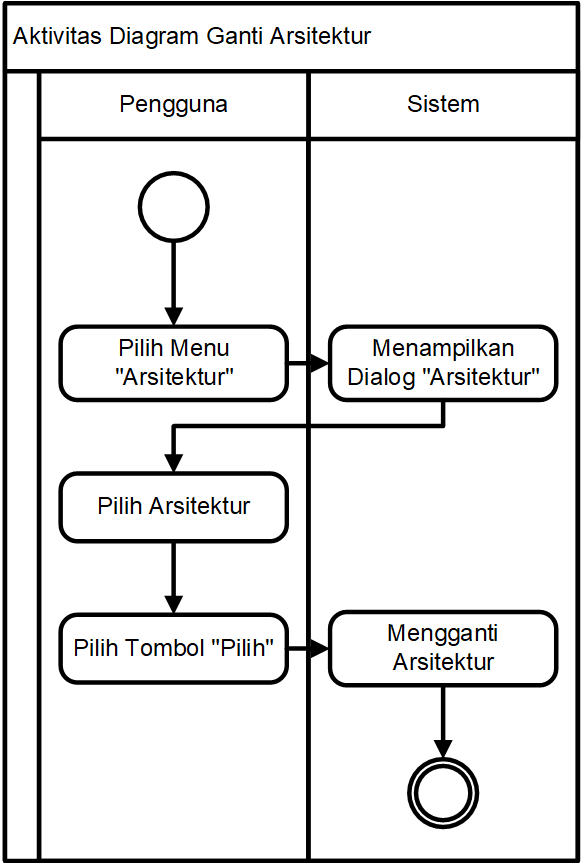
Adapun *about activity diagram* dapat dilihat pada gambar 3. 19.



Gambar 3. 19 *About Activity Diagram*

1. *Change Architecture* *Activity* *Diagram*

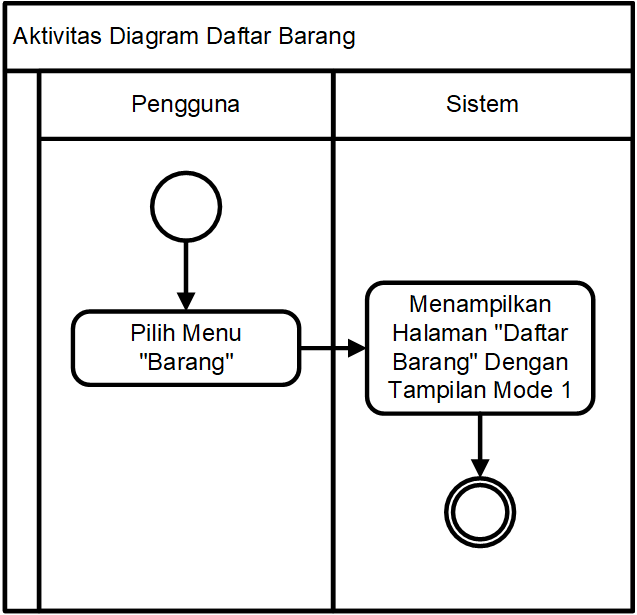
Adapun *change architecture activity diagram* dapat dilihat pada gambar 3. 20.



Gambar 3. 20 *Change Architecture Activity Diagram*

1. *List of Items Activity* *Diagram*

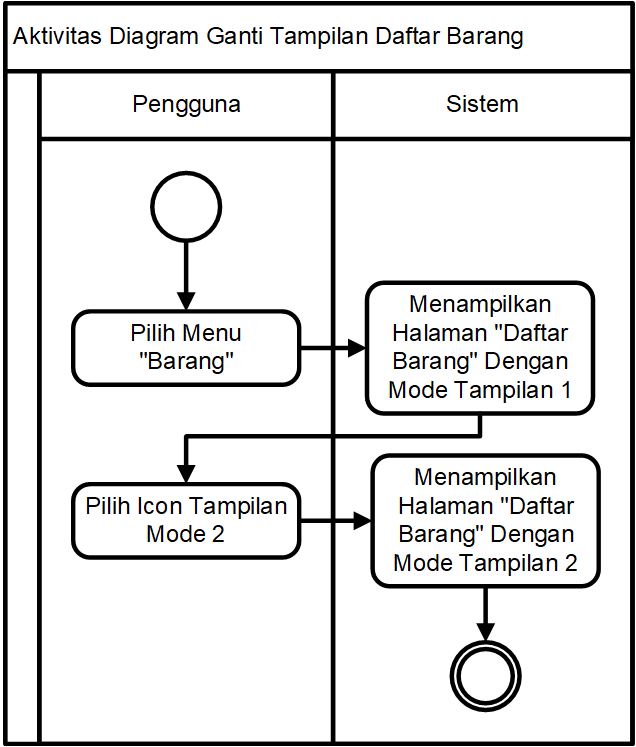
Adapun *list of items activity diagram* dapat dilihat pada gambar 3. 21.



Gambar 3. 21 *List of Items Activity Diagram*

1. *Change the Item List View* *Activity Diagram*

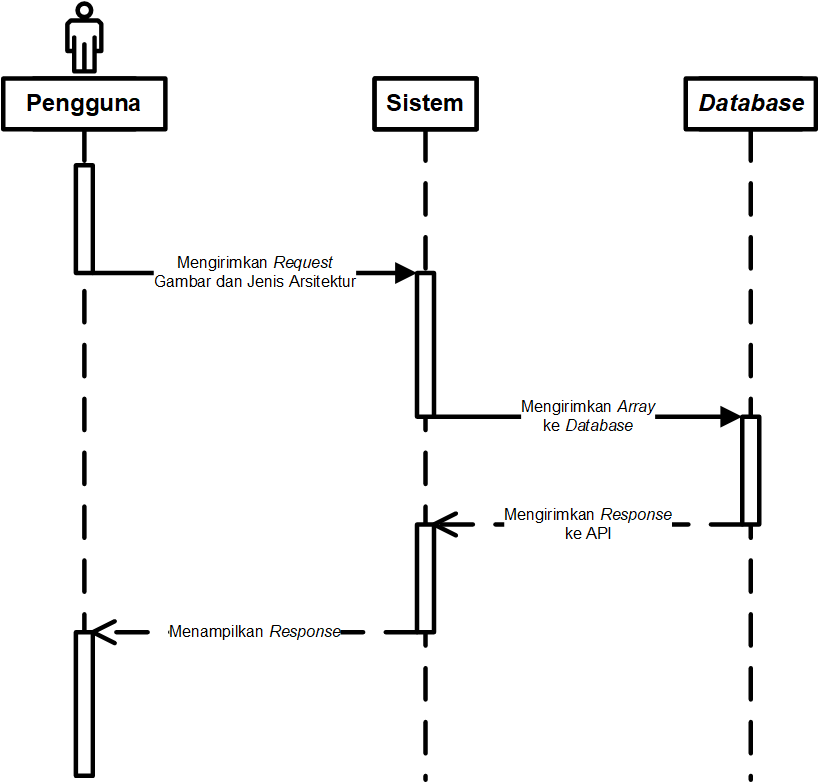
Adapun *change the item list view activity diagram* dapat dilihat pada gambar 3. 22.



Gambar 3. 22 *Change the Item List View* *Activity Diagram*

1. *Sequence* *Diagram*
2. *Post Picture Sequence* *Diagram*

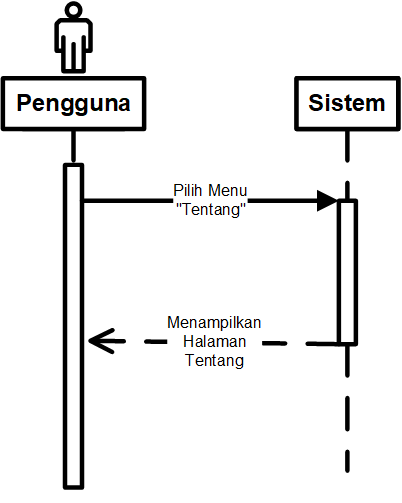
Adapun *post picture sequence diagram* dapat dilihat pada gambar 3. 23.



Gambar 3. 23 *Post Picture Sequence Diagram*

1. *About Sequence* *Diagram*

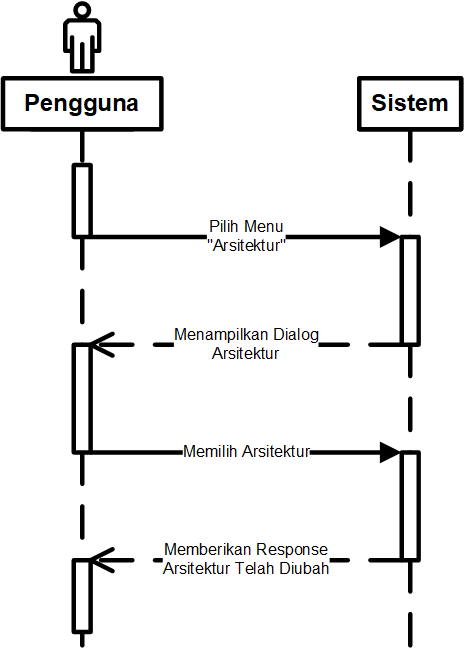
Adapun *about sequence diagram* dapat dilihat pada gambar 3. 24.



Gambar 3. 24 *About Sequence Diagram*

1. *Change Architecture* *Sequence Diagram*

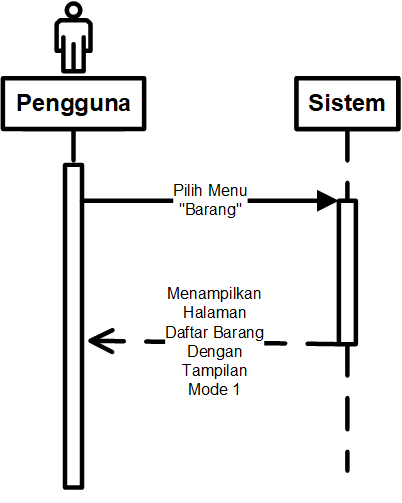
Adapun *change architecture sequence diagram* dapat dilihat pada gambar 3. 25.



Gambar 3. 25 *About Sequence Diagram*

1. *List of Items Sequence* *Diagram*

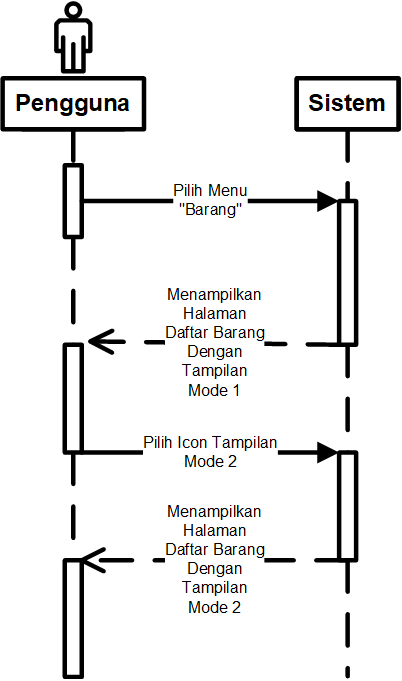
Adapun *list of items sequence diagram* dapat dilihat pada gambar 3. 26.



Gambar 3. 26 *List of Items Sequence Diagram*

1. *Change the Item List View Sequence* *Diagram*

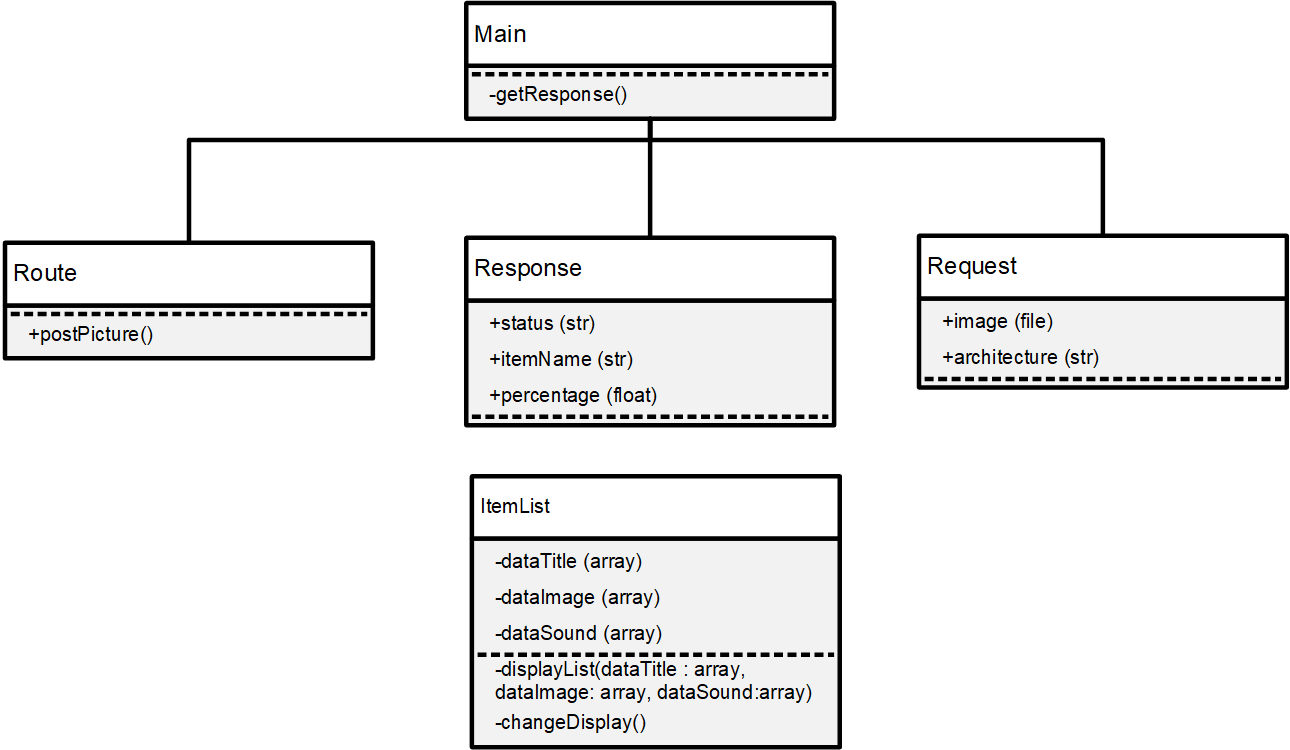
Adapun *change the item list view sequence* *diagram* dapat dilihat pada gambar 3. 27.



Gambar 3. 27 *Change the Item List View Sequence Diagram*

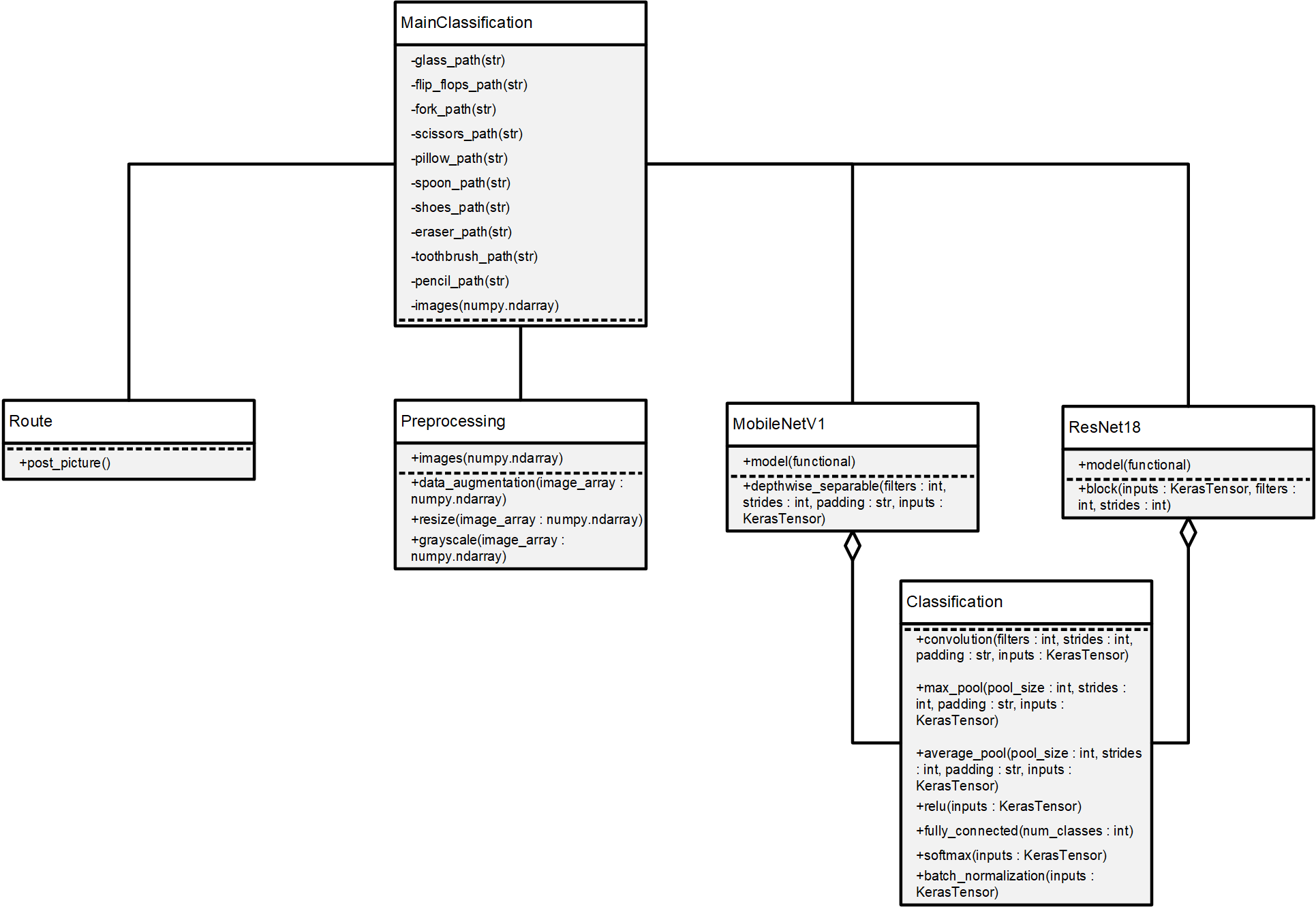
1. *Class* *Diagram*

Penelitian ini menggunakan dua sistem yaitu *Android* sebagai sarana pengguna untuk mengambil gambar dan API sebagai tempat pemrosesan klasifikasi gambar. Adapun *class diagram* untuk *Android* dapat dilihat pada gambar 3. 28.



Gambar 3. 28 *Android Class Diagram*

Adapun *class diagram* untuk API dapat dilihat pada gambar 3. 29.



Gambar 3. 29 *API Class Diagram*

## Analisis Perancangan *Interface*

Desain *interface* merupakan antarmuka yang menghubungkan antara user dengan sistem. Setelah melakukan proses analisis terhadap perancangan sistem, maka dilakukan perancangan *interface* dari aplikasi pengenal barang rumah ini. Adapun rancangan *interface* dari sistem ini adalah sebagai berikut:

1. Halaman *Splash Screen*

Halaman ini digunakan untuk memberikan informasi pendahuluan kepada pengguna saat mereka memulai aplikasi. Halaman ini akan tampil selama 2 detik dan akan langsung masuk ke halaman *dashboard*.



Gambar 3. 30 Halaman *Splash Screen*

1. Halaman *Dashboard*

Halaman ini digunakan untuk pengguna mengambil gambar dan dapat melihat hasil klasifikasi dari gambar tersebut.



Gambar 3. 31 Halaman *Dashboard*

1. Menu

Dalam aplikasi ini terdapat 3 menu, yaitu arsitektur, barang dan tentang. Menu arsitektur akan mengarahkan ke dialog untuk mengganti jenis arsitektur. Menu barang akan mengarahkan pengguna ke halaman daftar barang. Menu tentang akan mengarahkan pengguna ke halaman tentang.



Gambar 3. 32 Menu

1. Dialog Arsitektur

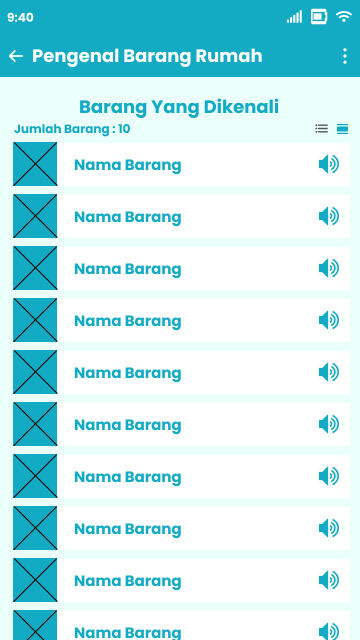
Dialog arsitektur merupakan dialog yang dapat digunakan pengguna untuk mengganti jenis arsitektur yang akan digunakan dalam klasifikasi gambar.



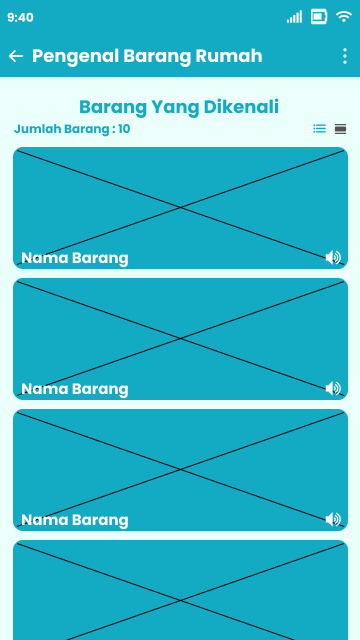
Gambar 3. 33 Dialog Arsitektur

1. Halaman Daftar Barang

Halaman ini digunakan untuk memberikan informasi kepada pengguna mengenai barang yang dapat dikenali dalam aplikasi ini beserta jumlahnya. Halaman ini memiliki dua mode tampilan.



Gambar 3. 34 Halaman Daftar Barang Mode 1



Gambar 3. 35 Halaman Daftar Barang Mode 2

1. Halaman *About*

Halaman ini digunakan untuk memberikan informasi kepada pengguna mengenai deskripsi aplikasi beserta profil pengembang aplikasi.



Gambar 3. 36 Halaman *About*

## Pengujian *Black Box*

Pengujian perangkat lunak digunakan untuk mengetahui apakah perangkat lunak dapat berjalan dengan baik atau tidak. Salah satu pengujian perangkat lunak yang digunakan dalam penelitian ini adalah *black box*. *Black box testing* digunakan untuk mengetahui apakah program yang dibuat sudah sesuai dengan kebutuhan fungsionalnya atau tidak.

Tabel 3. 4 Pengujian *Black Box System*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Halaman yang Diuji** | **Aksi** | **Reaksi Sistem Yang Diharapkan** |
| 1. | *Splash Screen* | Membuka aplikasi | *Splash Screen* tampil ± 2 detik, lalu masuk ke halaman *dashboard* |
| 2. | *Dashboard* | Menekan tombol “Ambil” | Kamera mengambil gambar yang ada di depannya dan menampilkannya di halaman *dashboard* |

Tabel 3. 4 Pengujian *Black Box System* (Lanjutan)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Halaman yang Diuji** | **Aksi** | **Reaksi Sistem Yang Diharapkan** |
| 2. | *Dashboard* | Menekan tombol “Ambil” | Hasil klasifikasi ditampilkan di halaman *dashboard* |
| Menekan gambar “Speaker” | Aplikasi membunyikan hasil klasifikasi melalui suara |
| Menekan menu “Arsitektur” | Aplikasi menampilkan dialog arsitektur |
| Menekan menu “Barang” | Aplikasi menampilkan halaman daftar barang |
| Menekan menu “Tentang” | Aplikasi menampilkan halaman tentang |
| 3. | Dialog Arsitektur | Memilih *radio button* “Residual Network 18” dan menekan tombol “Pilih” | Aplikasi mengganti jenis arsitektur menjadi *Residual Network 18* |
| Memilih *radio button* “Mobile Network V1” dan menekan tombol “Pilih” | Aplikasi mengganti jenis arsitektur menjadi *Mobile Network V1* |
| 4. | *About* | Menekan gambar “Github” | Aplikasi mengarahkan ke profil *github* pengembang |
| Menekan gambar “Instagram” | Aplikasi mengarahkan ke profil *instagram* pengembang |
| Menekan gambar “LinkedIn” | Aplikasi mengarahkan ke profil *linkedin* pengembang |

Tabel 3. 4 Pengujian *Black Box System* (Lanjutan)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Halaman yang Diuji** | **Aksi** | **Reaksi Sistem Yang Diharapkan** |
| 5. | Daftar Gambar | Menekan gambar “Mode Tampilan” | Aplikasi mengganti mode tampilan daftar barang |
| Menekan gambar “Speaker” | Aplikasi membunyikan nama barang melalui suara |

## Pengujian *Confusion Matrix*

Pengujian *confusion matrix* digunakan untuk mengukur kemampuan dari dua arsitektur CNN, yaitu *Residual Network 18* dan *Mobile Network V1*. Contoh pengujian *confusion matrix* yang akan dilakukan dalam penelitian ini dapat dilihat di bawah ini.

Sebuah sistem klasifikasi barang melakukan pengujian *confusion matrix*. Sistem ini memiliki 10 kategori barang dan dilakukan pengujian menggunakan 100 barang. Dalam pengujian sistem, terdapat 60 barang yang termasuk dalam salah satu dari kategori barang dan diklasifikasikan dengan benar, 20 barang yang bukan termasuk dalam salah satu dari kategori barang tetapi diklasifikasikan sebagai salah satu dari kategori tersebut, 15 barang yang bukan termasuk dalam salah satu dari kategori barang dan diklasifikasikan sebagai bukan salah satu dari kategori tersebut dan 5 barang yang termasuk salah satu dari kategori barang tetapi tidak diklasifikasi sebagai salah satu dari kategori tersebut.

Tabel 3. 5 Pengujian Confusion Matrix

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Prediksi | |
| Positif | Negatif |
| Aktual | Positif | 60 | 15 |
| Negatif | 20 | 5 |

Dalam kasus di atas, 60 barang dikategorikan sebagai *True Positive*, 20 barang dikategorikan sebagai *False Positive*, 15 barang dikategorikan sebagai *False Negative* dan 5 barang dikategorikan sebagai *True Negative*.

|  |  |
| --- | --- |
| a. |  |
|  |  |
| b. |  |
|  |  |
| c. |  |
|  |  |
| d. |  |

# DAFTAR PUSTAKA

Achmad, B. and Firadusy, K., 2013, *Pengolahan Citra Digital Menggunakan Delphi*. 1st edn, Yogyakarta: Andi.

Agrawal, S.A., Rewaskar, V.D., Agrawal, R.A., Chaudhari, S.S., Patil, Y. and Agrawal, N.S., 2023, Advancements in NSFW Content Detection: A Comprehensive Review of ResNet-50 Based Approaches, *International Journal of Intelligent Systems and Applications in Engineering*, pp. 41–45.

Ahdan, S., Pambudi, T., Sucipto, A. and Nurhuda, Y.A., 2019, Game untuk Menstimulasi Kecerdasan Majemuk pada Anak (Multiple Intelligence) Berbasis Android, in *Seminar Nasional Teknik Elektro UIN Sunan Gunung Djati Bandung (SENTER 2019)* , Bandar Lampung: Universitas Teknokrat Indonesia, pp. 554–568.

Anardani, S. and Putera, A.R., 2019, Analisis Pengujian Sistem Informasi Website E-Commerce Manies Group Menggunakan Metode Black Box Functional Testing  , in *Seminar Nasional Hasil Penelitian dan Pengabdian kepada Masyarakat UNIPMA*, Madiun: Universitas PGRI Madiun, pp. 72–75.

Anhar and Putra, R.A., 2023, Perancangan dan Implementasi Self-Checkout System pada Toko Ritel menggunakan Convolutional Neural Network (CNN), *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 11, pp. 466–478.

Cheah, K.H., Nisar, H., Yap, V.V., Lee, C.-Y. and Sinha, G.R., 2021, Optimizing Residual Networks and VGG for Classification of EEG Signals: Identifying Ideal Channels for Emotion Recognition, *Hindawi Journal of Healthcare Engineering*, pp. 1–14.

Community Refugee Sponsors Australia, 2022, Furniture and Basic Household Goods, *Community Refugee Sponsors Australia*, 1 October, pp. 1–3.

Cun, Y. Le, Boser, B., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D., 1990, Handwritten Digit Recognition with a Back-Propagation Network, pp. 396–404.

Direktorat Statistik Kependudukan dan Ketenagakerjaan, 2023, *Keadaan Pekerja di Indonesia Feruari 2023*, Jakarta: Badan Pusat Statistik Indonesia.

eWolf Community, 2012, *Index Lengkap Syntax*, Yogyakarta: MediaKom.

Fadjeri, A., Saputra, B.A., Ariyanto, D.K.A. and Kurniatin, L., 2022, Karakteristik Morfologi Tanaman Selada Menggunakan Pengolahan Citra Digital, *Jurnal Ilmiah Sinus (JIS)*, 20, pp. 1–12.

Frick, H. and Mulyani, T.H., 2006, *Arsitektur Ekologis*, Yogyakarta: Kanisius.

Fukushima, K., 1980, Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position, *Biological Cybernetics*, pp. 193–202.

Furusho, Y. and Ikeda, K., 2019, ResNet and Batch Normalization Improve Data Separability, in W.S. Lee and T. Suzuki (eds) *Proceedings of Machine Learning Research*, Nagoya: Asian Conference on Machine Learning, pp. 94–108.

He, K., Zhang, X., Ren, S. and Sun, J., 2015, Deep Residual Learning for Image Recognition.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017, MobileNets Efﬁcient Convolutional Neural Networks for Mobile Vision Applications, pp. 1–9.

Kementerian Pendidikan dan Kebudayaan Republik Indonesia, 2014, *Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 137 Tahun 2014 Tentang Standar Nasional Pendidikan Anak Usia Dini*, Jakarta: Kementerian Pendidikan dan Kebudayaan Republik Indonesia.

Kementerian Pendidikan, K.R. dan T., 2021, *Panduan APE Bagi Anak Usia Dini*, Jakarta: Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi.

Khotimah, S.H., Sunaryati, T. and Suhartini, S., 2021, Penerapan Media Gambar Sebagai Upaya dalam Peningkatan Konsentrasi Belajar Anak Usia Dini, *Jurnal Obsesi : Jurnal Pendidikan Anak Usia Dini*, 5(1), pp. 676–685.

Loekman, F. and Lina, 2023, Sistem Manajemen Inventori Dengan Pengenalan Barang Secara Otomatis Menggunakan Metode Convolutional Neural Network, *TEKNIKA*, 12, pp. 47–56.

Mahmud, K.H., Adiwijaya and Faraby, S. Al, 2019, Klasiﬁkasi Citra Multi-Kelas Menggunakan Convolutional Neural Network, *e-Proceeding of Engineerin*, 6, pp. 2127–2136.

Manurung, M. and Rahardja, P., 2008, *Pengantar Ilmu Ekonomi (Mikroekonomi & Makroekonomi)*. 3rd edn, Jakarta: Fakultas Ekonomi Universitas Indonesia.

Maulana, D.F., 2013, *Analisis dan Implementasi Penggunaan Metode Interpolasi Nearest Neighbor dan Interpolasi Bilinear (Studi Kasus Aplikasi Image Viewer Berbasis Android)*, Yogyakarta: Naskah Publikas AMIKOM Yogyakarta.

Musfiroh, D., Khaira, U., Utomo, P.E.P. and Suratno, T., 2021, Analisis Sentimen terhadap Perkuliahan Daring di Indonesia dari Twitter Dataset Menggunakan InSet Lexicon, *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 1, pp. 24–33.

Norma, 2017, *Pembesaran Citra Digital Berformat Bitmap Menggunakan Metode Nearest Neighbour*, Samarinda: Politeknik Negeri Samarinda.

Nurhidayat, A. and Esye, Y., 2022, Analisis Lampu Penerangan Rumah Tinggal Berdasarkan Lux dan Intensitas Konsumsi Energi, *Jurnal Sains & Teknologi Fakultas Teknik Universitas Darma Persada* [Preprint].

P, I.W.S.E., Wijaya, A.Y. and Soelaiman, R., 2016, Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101, *JURNAL TEKNIK ITS*, 5, pp. 65–69.

Pakaya, R., Tapate, A.R. and Suleman, S., 2020, Perancangan Aplikasi Penjualan Hewan Ternak Untuk Qurban dan Aqiqah dengan Metode Unified Modeling Language (UML) , *JTech*, pp. 31–40.

Paul, M.R., 2022, Classification of Garments from Fashion MNIST Dataset Using AlexNet CNN Architecture, *EPRA International Journal of Multidisciplinary Research (IJMR)*, 8, pp. 296–299.

Pemerintah Indonesia, 1992, *Undang-Undang Republik Indonesia Nomor 4 Tahun 1992 Tentang Perumahan dan Permukiman*, Jakarta: Sekretariat Negara.

Pemerintah Indonesia, 2002, *Undang-Undang Republik Indonesia Nomor 23 Tahun 2002 Tentang Perlindungan Anak*, Jakarta: Sekretariat Negara.

Pemerintah Indonesia, 2003, *Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 Tentang Sistem Pendidikan Nasional*, Jakarta: Sekretariat Negara.

Prabowo, D.A., Abdullah, D. and Manik, A., 2018, Deteksi dan Perhitungan Objek Berdasarkan Warna Menggunakan Color Object Tracking , *Jurnal Pseudocode*, 5, pp. 85–91.

Putri, F.G. and Andjarwirawan, J., 2022, Penerapan Metode Convolutional Neural Network Untuk Clothing Image Recognition, *JURNAL INFRA*, 10.

Rachmat, I.F.M., 2022, Pengembangan Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja Dengan Metode Rational Unified Process (Studi Kasus Klinik Argaraya Medika Malang) , *Jurnal Pengembangan Teknologin Informasi dan Ilmu Komputer*, 10, pp. 34–43.

Sandag, G.A. and Waworundeng, J., 2021, Identifikasi Foto Fashion Dengan Menggunakan Convolutional Neural Network (CNN) , *Cogito Smart Journal*, 7, pp. 305–314.

Singh, M., Verma, A., Parasher, A., Chauhan, N. and Budhiraja, G., 2019, Implementation of Database Using Python Flask Framework, *International Journal Of Engineering And Computer Science*, 8, pp. 24894–24899.

Suijono, Y.N., 2014, *Metode Pengembangan Kognitif*, Jakarta: Universitas Terbuka.

Sukamto, R.A. and Shalahudin, M., 2018, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, Bandung: Informatika Bandung.

Susilawati, F.E., Suparman and Patalo, A.M.G., 2022, Pengujian Black Box Aplikasi Penjualan Pupuk Bersubsidi Menggunakan Teknik Equivalence Partitioning, *Jurnal Ilmiah Information Technology d’Computare*, 12, pp. 30–34.

Toghyani, Shorabi, S.F., Shorabi, S.H. and Tabrizi, G., 2015, Check The Status Of The Development Of Children Underage 5 In Rural Areas Of Isfahan Using The ASQ Questionnaire In 2012-2013 Year, *Journal of Medicine and Life*, 8(4), pp. 169–173.

Vania, S., Sutanto, P., Sutanto, R. and Santoso, J., 2023, Ekstraksi Partitur Balok Monofonik Untuk Instrumen Flute dengan CRNN dan CRF, *Journal of Intelligent System and Computation*, 5, pp. 4–9.

Yang, M., McGrath, R.E. and Folk, M., 2005, HDF5 – A High Performance Data Format for Earth Science, in *Proceeding of the International Conference on Interactive Information Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*, Urbana-Champaign: National Center for Supercomputing Applications, pp. 1–6.